1

Notes on the Space Requirements for Checking Satisfiability in Modal Logics

MARCUS KRACHT

ABSTRACT. Recently, there has been growing attention to the space requirements of tableau methods (see for example [7], [1], [12]). We have proposed in [10] a method of reducing modal consequence relations to the global and local consequence relation of (polymodal) K. The reductions used there did however not establish good time complexity bounds. In this note we shall use reduction functions to obtain rather sharp space bounds. These bounds can also be applied to show completeness of ordinary tableau systems, which in turn yield space bounds that are slightly different from the ones derived by applying the reduction functions alone.

It has been shown by Hudelmaier ([7]) that satisfiability in $\mathsf{S4}$ is $O(n^2 \log n)$–space computable, while satisfiability in K and KT are $O(n \log n)$–space computable. An $O(n \log n)$–space bound for KD has been obtained by Basin, Matthews and Viganò ([1]). Viganò ([15]) has shown that satisfiability in $\mathsf{K4}$, $\mathsf{KD4}$ and $\mathsf{S4}$ is in $O(n^2 \log n)$–space. Nguyen has reduced these bounds to $O(n \log n)$ for $\mathsf{K4}$, $\mathsf{K4D}$ and $\mathsf{S4}$ in [12]. The bound for K has been improved to $O(n)$ by Hemaspaandra in [5].

We shall deal here with an abstract method for obtaining space bounds, using reduction functions. Reduction functions have been introduced in [9] and further developed in [10]. These functions were used to obtain a number of folklore results on standard modal systems in a uniform way. Examples were the finite model property and interpolation, but also complexity. However, the bounds for local satisfiability obtained there were not good enough. Here we shall improve these results rather drastically. In some cases we shall obtain linear bounds (namely for $\mathsf{K.D}$, $\mathsf{K.alt_1}$), while in other cases we shall obtain a bound which is a product of the number of subformulae and the modal depth. In some cases this is somewhat less than the best known result (it can be quadratic in the length), in others it is better. Typically, with the help of tableau methods we can also establish the $O(n \log n)$ space bound for most systems.

The research for this paper has been made possible by a Heisenberg–grant by the Deutsche Forschungsgemeinschaft. I wish to express my thanks to Rajeev Goré, Edith Hemaspaandra, Fabio Massaci and two anonymous referees for useful discussions. Special thanks go to Frank Wolter for his help. The responsibility for errors and omissions is solely mine.

1 Preliminaries

We refer to [9] for details concerning modal logic. The language consists of the set $\{p_i : i \in \omega\}$ of variables and the symbols $\bot$, $\neg$, $\wedge$ and, finally, the modal operators $\Box_i$, $i < q$. (The letter 'q' will be reserved throughout this paper for the number of basic operators.) The symbols $\vee$ and $\to$ are used here, but they are only abbreviations. For sets of formulae we use the following shorthand notation. For a formula φ, $\mathrm{dg}(\varphi)$ is the maximum nesting of modal operators. It is called the **modal degree of** φ. For a finite set Δ, $\mathrm{dg}(\Delta)$ is the maximum of the modal degrees of its members.

As usual, $\varphi; \psi$ denotes $\{\varphi, \psi\}$, $\Delta; \varphi$ denotes $\Delta \cup \{\varphi\}$ and $\Delta; \Theta$ denotes $\Delta \cup \Theta$. Thus $\Delta; \varphi; \varphi$ is the same set as $\Delta; \varphi$. For a formula φ, $\mathrm{sf}(\varphi)$ denotes the set of subformulae of φ, $\mathrm{sf}^{\neg}(\varphi) := \mathrm{sf}(\varphi) \cup \{\neg\delta : \delta \in \mathrm{sf}(\varphi)\}$, $\mathrm{var}(\varphi)$ the set of variables occurring in φ. For a set Δ of formulae we put

$$\mathrm{sf}(\Delta) := \bigcup\{\mathrm{sf}(\delta) : \delta \in \Delta\} \tag{1}$$

$$\mathrm{sf}^{\neg}(\Delta) := \bigcup\{\mathrm{sf}^{\neg}(\delta) : \delta \in \Delta\} \tag{2}$$

$$\mathrm{var}(\Delta) := \bigcup\{\mathrm{var}(\delta) : \delta \in \Delta\} \tag{3}$$

A modal logic is identified here with its set of theorems. The least monomodal logic is called K. With q the number of basic operators, the least q–modal logic is denoted by K_q. (So, K_0 is PC, and $\mathsf{K}_1 = \mathsf{K}$.) With a modal logic L, two consequence relations are associated: $\vdash_L$ and $\Vdash_L$. The first is called the **local consequence relation** and is defined as follows. $\Delta \vdash_L \varphi$ iff φ can be proved from $\Delta \cup L$ using only Modus Ponens. The second is called the **global consequence relation** and defined as follows. $\Delta \Vdash_L \varphi$ iff φ can be derived from $\Delta \cup L$ using Modus Ponens and Necessitation.

Problems are functions $f : A^* \to \{\mathtt{0}, \mathtt{1}\}$, where A is a finite set of so–called **letters**. We shall study the problems '$\Delta \vdash_L \varphi$' and '$\Delta \Vdash_L \varphi$' for various logics L, which are the following. Given a finite set Δ and a formula φ, determine whether $\Delta \vdash_L \varphi$ ($\Delta \Vdash_L \varphi$) and if so, output $\mathtt{1}$, and if not, output $\mathtt{0}$. Given a complexity class $\mathcal{C}$, a problem f is said to be **in** $\mathcal{C}$ if f is computable in $\mathcal{C}$; f is $\mathcal{C}$**–hard**, if for any problem g in $\mathcal{C}$ there is a function h in $\mathcal{C}$ such that $f \circ h = g$; and f is $\mathcal{C}$**–complete** if it is both in $\mathcal{C}$ and $\mathcal{C}$–hard. We assume that $\mathcal{C}$ is closed under composition.

Definition 1.1 *Let L be a modal logic and $\mathcal{C}$ a complexity class. We say that L is **locally** in $\mathcal{C}$ (is **locally** $\mathcal{C}$–**hard**, is **locally** $\mathcal{C}$–**complete**) if the* *problem '$\Delta \vdash_L \varphi$' is in $\mathcal{C}$ (is $\mathcal{C}$–hard, is $\mathcal{C}$–complete). Likewise, L is* ***globally*** *in $\mathcal{C}$ (is* ***globally*** *$\mathcal{C}$–**hard**, is* ***globally*** *$\mathcal{C}$–**complete**) if the problem '$\Delta \Vdash_L \varphi$' is in $\mathcal{C}$ (is $\mathcal{C}$–hard, is $\mathcal{C}$–complete).*

Recall the following. First, let $\Box^k\varphi$ be defined by induction as usual.

$$\Box^{<1}\varphi := \varphi \tag{4}$$

$$\Box^{<k+1}\varphi := \varphi \wedge \Box\Box^{<k}\varphi \tag{5}$$

Also,

$$\Box^k\Delta := \{\Box^k\delta : \delta \in \Delta\} \tag{6}$$

$$\Box^{<k}\Delta := \{\Box^{<k}\delta : \delta \in \Delta\} \tag{7}$$

Lemma 1.2 *(a) $\Delta \Vdash_L \varphi$ iff for some k: $\Box^{<k}\Delta \vdash_L \varphi$. (b) $\Vdash_L \varphi$ iff $\vdash_L \varphi$.*

Given a finite set Δ and a formula φ, we have $\Delta \vdash_L \varphi$ iff $\vdash_L \bigwedge\Delta \to \varphi$ iff $\Vdash_L \bigwedge\Delta \to \varphi$. Thus, local derivability problems are equivalent to local satisfiability problems, which are subproblems of global derivability problems. (Moreover, the transformation is log–space computable.)

Proposition 1.3 *Let $\mathcal{C}$ be a complexity class and L a modal logic. If L is locally $\mathcal{C}$–hard, it is also globally $\mathcal{C}$–hard. If it is globally in $\mathcal{C}$, it is also locally in $\mathcal{C}$.*

Since satisfiability in consistent modal logics is at least NP–hard, the complexity of satisfiability in modal logics does not change under linear reductions. (For the inconsistent logic the satisfiability and derivability problems are of course trivial.)

2 Reduction Functions

Definition 2.1 (Local Reduction Function) *Let $L \subseteq M$ be two logics. Let Y be a function defined on the set of finite sets of formulae, with range the set of finite sets of formulae. Y is called a* ***local reduction function*** *from M to L if $\Delta \vdash_M \varphi$ iff (1) $\Delta; Y(\Delta;\varphi) \vdash_L \varphi$, and (2) for all Δ $\mathrm{var}(Y(\Delta)) \subseteq \mathrm{var}(\Delta)$.*

The condition on variables will not be needed here but is essential in proofs concerning interpolation (see [9]).

Definition 2.2 (Global Reduction Function) *Let $L \subseteq M$ be two logics. Let X be a function defined on the set of finite sets of formulae, with range the set of finite sets of formulae. X is called a* ***global reduction***

function *from M to L if $\Delta \Vdash_M \varphi$ iff (1) $\Delta; X(\Delta;\varphi) \Vdash_L \varphi$ and (2) for all Δ $\mathrm{var}(X(\Delta)) \subseteq \mathrm{var}(\Delta)$.*

These definitions are as given in [10], where the following were established to be global reduction functions:

$$X_{\mathsf{4}}(\Delta) := \{\Box\chi \to \Box\Box\chi : \Box\chi \in \mathrm{sf}(\Delta)\} \tag{8}$$

$$X_{\mathsf{T}}(\Delta) := \{\Box\chi \to \chi : \Box\chi \in \mathrm{sf}(\Delta)\} \tag{9}$$

$$X_{\mathsf{B}}(\Delta) := \{\neg\chi \to \Box\neg\Box\chi : \Box\chi \in \mathrm{sf}(\Delta)\} \tag{10}$$

$$X_{\mathsf{G}}(\Delta) := \{\neg\Box\chi \to \neg\Box(\chi \vee \neg\Box\chi) : \Box\chi \in \mathrm{sf}(\Delta)\} \tag{11}$$

$$X_{\mathsf{Grz}}(\Delta) := \{\neg\Box\chi \to \neg\Box(\chi \vee \neg\Box(\chi \to \Box\chi)) : \Box\chi \in \mathrm{sf}(\Delta)\} \tag{12}$$

$$X_{\mathsf{alt}_1}(\Delta) := \{\neg\Box\chi \to \Box\neg\chi : \Box\chi \in \mathrm{sf}(\Delta)\} \tag{13}$$

Additional reduction functions for $\mathsf{K.D}$ and $\mathsf{K.5}$ are

$$X_{\mathsf{D}}(\Delta) := \{\neg\Box\bot\} \tag{14}$$

$$X_{\mathsf{5}}(\Delta) := \{\neg\Box\chi \to \Box\neg\Box\chi : \Box\chi \in \mathrm{sf}(\Delta)\} \tag{15}$$

The functions X_{G}, X_{5}, are reduction functions to $\mathsf{K4}$, X_{Grz} is a reduction to $\mathsf{S4}$. All others are to K.

The results of this paper could be slightly improved if we define a function on the basis of the pair $\langle\Delta,\varphi\rangle$ rather than the set $\Delta;\varphi$. However, this gives no improvement on the theoretical complexity of the algorithms. Hence we have ignored this finesse here.

3 Local and Global Tableaux

We shall first sketch a method for obtaining space bounds for (polymodal) K. The first calculus, basically folklore (see [13]), shall be called the **local tableau calculus**, to distinguish it from the second, the so–called **global tableau calculus**.

The calculi as presented below operate on *sets* of formulae, while they should be thought of as operating on strings. This is just a matter of presentation. The space estimates will be always given in terms of the input string, which is an ordered sequence of formulae. For the purpose of the calculus, let $\Delta_{\Box_j} := \{\chi : \Box_j\chi \in \Delta\}$. The rules are as follows.

$$(\neg\mathrm{E})\quad \frac{\Delta;\neg\neg\varphi}{\Delta;\varphi} \qquad\qquad (\wedge\mathrm{E})\quad \frac{\Delta;\varphi\wedge\psi}{\Delta;\varphi;\psi}$$

$$(\vee\mathrm{E})\quad \frac{\Delta;\neg(\varphi\wedge\psi)}{\Delta;\neg\varphi|\Delta;\neg\psi} \qquad\qquad (\Box_j\mathrm{E})\quad \frac{\Delta;\neg\Box_j\varphi}{\Delta_{\Box_j};\neg\varphi}$$

A **tableau** for Δ is a maximally 2–branching tree with nodes labeled by finite sets of formulae and a name of a rule (except for leaves) such that the root is labeled Δ, and the following holds.

1. If x has label ($\neg$E), ($\wedge$E) or ($\Box_j$E), then it has a unique daughter and the daughter carries the appropriate label.
2. If x has label ($\vee$E) then it has exactly two daughters, one for each set mentioned by the rule.

A tableau is **closed** if all leaves carry a Δ such that $\bot \in \Delta$ or both $\varphi \in \Delta$ and $\neg\varphi \in \Delta$ for some φ. Δ is **rejected** by the tableau calculus if it has a closed tableau. We remark without proof that this calculus is sound and complete for K in the sense that a set of formulae is rejected iff it is inconsistent. For the purpose of computing with tableaux we shall think of the sets of formulae as sequences possibly containing multiple occurrences of formulae. The semicolon is then to be thought of as sequence concatenation. It is possible to remove double occurrences of formulae (to save space). In this case we agree to keep the first occurrence of a formula in the sequence.

The local calculus is just part of the global calculus, which is as follows (see also [3]). It runs on pairs of sets of formulae, separated by †. We interpret $\Theta \dagger \Delta$ as a set of formulae of which the members of Θ hold globally and the members of Δ hold only locally. The rules are as follows.

$$(\neg\mathrm{E}) \quad \frac{\Theta \dagger \Delta; \neg\neg\varphi}{\Theta \dagger \Delta; \varphi} \qquad\qquad (\wedge\mathrm{E}) \quad \frac{\Theta \dagger \Delta; \varphi \wedge \psi}{\Theta \dagger \Delta; \varphi; \psi}$$

$$(\vee\mathrm{E}) \quad \frac{\Theta \dagger \Delta; \neg(\varphi \wedge \psi)}{\Theta \dagger \Delta; \neg\varphi | \Theta \dagger \Delta; \neg\psi} \qquad\qquad (\Box_j\mathrm{E}) \quad \frac{\Theta \dagger \Delta; \neg\Box_j\varphi}{\Theta \dagger \Theta; \Delta_{\Box_j}; \neg\varphi}$$

This calculus is also sound and complete in the following sense. $\Theta \dagger \Theta; \Delta$ has a closed tableau iff $\neg \bigwedge \Delta$ globally follows in K from Θ. This is easily seen. If we put $\Theta := \varnothing$ we obtain the local calculus.

Notice that both tableau calculi use a somewhat larger set than the set of subformulae. Namely, if the starting set is Δ (or $\Theta \dagger \Delta$), then the sets in the tableau remain within the set $\mathrm{sf}^{\neg}(\Delta)$ ($\mathrm{sf}^{\neg}(\Theta; \Delta)$).

4 Space Requirements of the Tableaux

We shall deal now with the space requirement for these tableau calculi.

Definition 4.1 (Syntax) *Let q be the number of basic modal operators. $A := \{\mathtt{p}, \mathtt{0}, \mathtt{1}, \wedge, \neg, \bot, \Box\}$, $S := \{;, \dagger\}$. Members of A are called **logical symbols**. The set of q–modal fomulae is a subset of A^*, which is defined as follows.*

1. *$\bot$ is a formula.*
2. *If $\alpha \in \{\mathtt{0}, \mathtt{1}\}^*$, then $\mathtt{p}\alpha$ is a variable. A variable is a formula.*
3. *If φ is a formula, so is $\neg\varphi$.*
4. *If φ is a formula, and α the binary code of a number $< q$ then $\Box\alpha\varphi$ is a formula.*
5. *If φ and χ are formulae, so is $\wedge\varphi\chi$.*

If $q = 1$, we use $\Box$ in place of $\Box\mathtt{0}$ in Item 4. Sequences of formulae (and pairs thereof) are elements of $(A \cup S)^*$, which are defined as follows.

1. *Every formula is a sequence of formulae.*
2. *If Δ and Δ' are sequences of formulae, so is $\Delta;\Delta'$.*
3. *If Θ and Δ are sequences of formulae, $\Theta \dagger \Delta$ is a pair of sequences of formulae.*

Unique readability can be shown for this notation. Notice that formulae are coded in Polish Notation, but quoted in running text in the usual way. The *length* of a formula, sequence or pair of sequences of formulae is its **length** as a string, denoted by $|\Delta|$. To eliminate dependency on variable names, one defines the **modified length**, $||\Delta||$, as follows.

1. $||\bot|| := ||\mathtt{p}\alpha|| := 1$.
2. $||\neg\varphi|| := ||\Box\varphi|| := ||\Box\alpha\varphi|| := ||\varphi|| + 1$.
3. $||\varphi \wedge \psi|| := ||\varphi|| + ||\psi|| + 1$.
4. $||\Delta;\Delta'|| := ||\Delta|| + ||\Delta'||$.

In general, $||\Delta|| \leq |\Delta|$. The most common length count is actually $||\Delta||$, which works however only if the number of operators is finite. (Otherwise, the length of α must be counted as well in the second clause above, but not in the first.) For details see [14].

It is worthwhile to explain here a measure of length that is used in [10]. Suppose that Δ is given. Then let $\sharp\Delta := \mathrm{card}(\mathrm{sf}(\Delta))$ be the number of subformulae of Δ. We remark here that $||\Box^{<k}\varphi|| = k(||\varphi|| + 1) - 1$ and $\sharp(\Box^{<k}\varphi) \leq (\sharp\varphi) + 2(k-1)$. For sets we have

$$||\Box^{<k}\Delta|| \leq k(||\Delta||) - \mathrm{card}(\Delta) \tag{16}$$

$$\sharp(\Box^{<k}\Delta) \leq 2(k-1)\,\mathrm{card}(\Delta) + \sharp\Delta \tag{17}$$

Most complexity results could be stated alternatively with this length function, though it is somewhat different from $|\Delta|$. Notice that the space needed to code Δ is not predictable from $\sharp\Delta$ alone, since Δ contains variable names of unpredictable length.

Proposition 4.2 $\log_2 ||\Delta|| \leq \sharp\Delta \leq ||\Delta||$.

None of the bounds can be improved. Let Δ be a set of formulae without $\bot$ in which every variable occurs at most once. Then $\sharp\Delta = ||\Delta||$. For the other inequality, define

$$\chi_0 := \mathtt{p0}, \qquad \chi_{n+1} := \wedge\chi_n\chi_n \tag{18}$$

χ_n has $n+1$ distinct subformulae and is of length $2^{n+1} + 2^{n-1} - 1$. Thus, $\log_2 ||\chi_n||$ is asymptotically equal to $\sharp\chi_n$. (This shows that $\log_2 ||\Delta|| \leq \sharp\Delta$ can in general not be improved.)

Computations considered here are implemented as computations over strings of formulae, which however represent sets of formulae. Since in sets we do not count repeated items, this leads to a slight improvement of the space complexity bounds.

Lemma 4.3 *A member of* $\mathrm{sf}(\Delta)$ *needs* $O(\log \sharp\Delta)$ *space to code.*

To that effect, think of Δ as written onto a tape. A set of subformulae of Δ can be coded by marking on Δ all positions where a subformula belonging to that set begins. This explains the bound $\log_2 |\Delta|$. However, notice that (a) there are positions where no subformula begins (so the bound may be $\log_2 ||\Delta||$), and (b) Δ may contain the same subformula several times (which reduces the bound to $\log_2 \sharp\Delta$). So we do the following. We use a second tape, where cell number j contains $\circ$ iff cell number j on the input tape begins a subformula, and moreover, no cell number $i < j$ on the input tape begins the same subformula. All other cells are marked by $\bullet$. Evidently, since the rules manipulate effectively only sets of subformulae, we need to deal exclusively with those cells marked $\circ$ in place of all cells. The auxiliary tape has length $|\Delta|$ but only $\sharp\Delta$ many cells marked $\circ$. Hence, the binary code of these cells (counted ignoring the $\bullet$ cells) consumes only $\log_2 \sharp\Delta$ space. To generate that tape, a few read heads are needed to scan the input. Hence we can eliminate the auxiliary tape by introducing several read heads on the input tape (equivalently, consuming additional $O(\log_2 |\Delta|)$ space). Since the initial sequence, Δ, is on the input tape, its space requirement is not counted for the space complexity.

A **binary tree domain** is a subset T of $\{\mathtt{0}, \mathtt{1}\}$ such that (a) if $\vec{s}^\frown j \in T$ then $\vec{s} \in T$, (b) if $\vec{s}^\frown\mathtt{1} \in T$ then $\vec{s}^\frown\mathtt{0} \in T$. A binary branching tree can be coded by a binary tree domain. A tableau can be coded as a set $\mathcal{T}$ of quadruples of the form $w = \langle \vec{s}, \rho, \vec{u}, \Delta \rangle$, where $\vec{s}$ is a binary sequence, ρ the name of a rule, $\vec{u}$ a binary sequence, and Δ a sequence of formulae. Moreover, the set of $\vec{s}$ such that there is $\langle \vec{s}, \rho, \vec{u}, \Delta \rangle$ shall form a tree domain, such that the following holds: for each $w = \langle \vec{s}^\frown j, \rho, \vec{u}, \Delta \rangle$ and $w' = \langle \vec{s}, \sigma, \vec{v}, \Delta' \rangle$ in $\mathcal{T}$, Δ is obtained from Δ' by applying rule ρ to the nth formula of Δ', where $\vec{u}$ is the binary code of n. Moreover, if $\rho = (\mathrm{E}\vee)$, the nth formula is $\neg(\varphi \wedge \psi)$ and $j = 0$, then $\neg\varphi$ is chosen to form Δ, and if $j = 1$, then $\neg\psi$ is chosen. (Notice that if $\vec{s} = \varepsilon$, the empty word, ρ and $\vec{u}$ can be anything, for example $\mathtt{0}$.) $\vec{s}$ is called the **name** of w, and Δ its **label**. The pair $\rho, \vec{u}$ is called the **transition code**. Notice immediately that ρ is redundant in presence of $\vec{u}$. If v has label $\vec{s}^\frown j$, $j \in \{\mathtt{0}, \mathtt{1}\}$, then v is called a **daughter of** w.

We shall first deal with the local tableau calculus.

Lemma 4.4 *Let Δ' be the label of a daughter of w in a local tableau, and Δ the label of w. Then $|\Delta'| \leq |\Delta|$. Moreover, $||\Delta'|| < ||\Delta||$.*

It follows that the length of a branch is linearly bounded in the number of logical symbols of Δ_0, the starting sequence. Now notice the following. Each daughter node is uniquely determined from its mother node by naming: (a) which formula the rule has been applied on, (b) whether we choose the left hand branch in case the formula is of the form $\neg(\varphi \wedge \psi)$, or the right hand branch. The information (b) is actually present in the name of the node and need not be given, since for the leftmost formula the node named $\vec{s}\mathtt{0}$ is chosen and for the rightmost formula the node named $\vec{s}\mathtt{1}$. All other information can be recomputed. So we may store the tableau as follows. Instead of $\langle \vec{s}, \rho, \vec{u}, \Delta \rangle$ for $\vec{s} \neq \varepsilon$ we only store $\langle \vec{s}, \vec{u} \rangle$. We have $|\vec{s}| \leq \log_2 ||\Delta||$, as we need to perform tableaux rules only once per occurrence of a subformula. Also, $|\vec{u}| \leq \log_2 |\Delta|$. By Lemma 4.3, $\vec{u}$ has length $\leq \log_2 \sharp\Delta$. Finally, if we code only branches, $\vec{s}$ can be reduced to its last member, so is in effect constant. So, a branch for Δ in a local tableau needs $O(||\Delta|| \log_2 \sharp\Delta)$ space to code.

Here is an algorithm that computes only a single branch at a time. Start with the tableau consisting of just one node and the opening sequence, and call x *unfinished.*

1. If x is unfinished and $(\neg\mathrm{E})$, $(\wedge\mathrm{E})$ or $(\vee\mathrm{E})$ is applicable on a formula then: x remains unfinished. Apply the rules once to the leftmost possible formula. In case of $(\vee\mathrm{E})$, choose the leftmost disjunct. This creates y. y is unfinished. Continue with y.

2. If x is unfinished, and only $(\Box\mathrm{E})$ is applicable at x then: apply $(\Box\mathrm{E})$

to the leftmost possible formula, create a successor y. y is unfinished. Continue with y.

3. If x is unfinished and no rule is applicable then: if it contains a formula and its negation, x becomes closed. If not, x becomes open. Continue with x.
4. If x is closed and has no parent node: exit. 'There is a closing tableau.'
5. If x is closed and has parent y:
   (a) If y is a ($\vee$E)–node and x is the left hand daughter of y then: create right hand daughter z. z is unfinished. Continue with z.
   (b) Else: y becomes closed. Continue with y.
6. If x is open and has no parent: exit. 'There is no closing tableau.'
7. If x is open and has parent y:
   (a) If y is a ($\vee$E)–node, a ($\wedge$E)–node or a ($\neg$E)–node then: y becomes open. Continue with y.
   (b) If y is a ($\Box$E)–node then: take the next suitable formula to apply ($\Box$E). If it does not exist, y becomes open. If it exists, y becomes unfinished. Continue with y.

Since branches are bounded in length by $||\Delta||$ we get the following.

Theorem 4.5 *It can be checked in $O(||\Delta|| \log_2 \sharp\Delta)$ space whether a given set Δ of fomulae is satisfiable in (polymodal) K.*

Since $\sharp\Delta \leq ||\Delta|| \leq |\Delta|$ this trivially implies that also $O(||\Delta|| \log_2 ||\Delta||)$ is also sufficient. Since complexity is typically measured in terms of $||\Delta||$, Theorem 4.5 is slightly better than the standard ones.

The tableau method can be applied also to other logics. For example, [13] has shown that if one adds the following rule

$$(19) \qquad (\Box\mathrm{T}) \quad \frac{\Delta; \Box\varphi}{\Delta; \varphi}$$

the resulting calculus is sound and complete for $\mathsf{K.T}$. (This can be proved also using reduction functions on the basis of the completeness of the calculus for K.) Evidently, no branch needs to be longer than $\sharp\Delta$, so we can restrict the space to $O(||\Delta|| \log \sharp\Delta)$.

Recently, Hemaspaandra [5] has observed that the space bounds can be improved even further. First, observe the following.

Lemma 4.6 *A subset of* $\mathrm{sf}(\Delta)$ *needs* $O(\sharp\Delta)$ *space to code.*

Namely, just note that a subset can be coded as a set of cells on the auxiliary tape (see the remarks following Lemma 4.3). Hemaspaandra steps directly from a downward saturated set to another downward saturated set using the following single rule:

$$(\Box\mathrm{H}) \qquad \frac{\Delta; \neg\Box_j\chi}{(\Delta_{\Box_j}; \neg\chi)^\star}$$

Here, $\Theta^\star$ denotes a saturated closure of Θ. This rule eliminates the rules ($\neg$E), ($\wedge$E) and ($\vee$E). A downward saturated set is simply a subset of $\sharp\Delta$. However, Hemaspaandra makes the following crucial observation. Let $\mathrm{sf}(d, \Delta)$ be the set of occurrences of subformulae that are exactly inside the scope of d many $\Box$. It is clear that $\mathrm{sf}(d, \Delta) \cap \mathrm{sf}(d', \Delta) = \varnothing$ whenever $d \neq d'$. (Think of occurrences as cells of the input string.) Now, if the set of occurrences above the line of ($\Box$H) was within $\mathrm{sf}^\neg(d, \Delta)$, then the set of occurrences below the line is within $\mathrm{sf}^\neg(d + 1, \Delta)$. Therefore the entire tableau can be coded inside $\mathrm{sf}^\neg(\Delta)$. Namely, a tableau is a sequence of the form $\langle\langle\Delta_i, \chi_i\rangle : i < n\rangle$, where Δ_i is a downward saturated subset of $\mathrm{sf}^\neg(i, \Delta)$ and $\chi_i \in \Delta_i$ the formula on which the rule is operated next. It is checked in linear space whether a set is downward saturated. The nondeterminism in choosing a successor set does not affect the space complexity. We can backtrack on the saturation (the saturated closures can be effectively enumerated), and we can backtrack on the formulae on which the rules have operated.

Now put

$$\sharp_\delta(\Delta) := \mathrm{card}(\bigcup_{n\in\omega} \mathrm{sf}(n, \Delta)) \tag{20}$$

(So, in $\sharp_\delta\Delta$ we count two occurrences of a subformula as different just in case their degree of embedding is different.) Hemaspaandra's result can be improved to $O(\sharp_\delta\Delta)$. It can be seen that $\sharp_\delta(\Delta) \leq \mathrm{dg}(\Delta)\sharp\Delta \leq (\sharp\Delta)^2$. Namely, the degree of embedding is bounded from above by the number of subformulae. Hence there are at most $\sharp\Delta$ occurrences of a subformula that have pairwise different degree of modal embedding. This bound can, however, not be improved. Define

$$\sigma_0 := \mathtt{p0}, \qquad \sigma_{n+1} := \wedge\sigma_n\Box\sigma_n \tag{21}$$

Then $\sharp\{\sigma_n\} = 2n+1$, $\mathrm{dg}(\sigma_n) = n$, $\sharp_\delta(\{\sigma_n\}) = (n+1)^2$. With badly designed formulae, therefore, Hemaspaandra's result gives us only $O((\sharp\Delta)^2)$. It is such type of formulae that occur a lot with the reduction functions.

In order to boost this up for the minimal q–modal logic K_q we just have to replace the number d by a sequence of numbers $< q$. Thus, we define the following sets. Let Δ be given. Then for each occurrence of a subformula we define the following.

1. For all $\delta \in \Delta$: $\delta \in \mathrm{sf}(\varepsilon, \Delta)$.
2. If $\neg\chi \in \mathrm{sf}(\vec{\alpha}, \Delta)$, then $\chi \in \mathrm{sf}(\vec{\alpha}, \Delta)$.
3. If $\varphi \wedge \chi \in \mathrm{sf}(\vec{\alpha}, \Delta)$, then $\varphi, \chi \in \mathrm{sf}(\vec{\alpha}, \Delta)$.
4. If $\Box_j \chi \in \mathrm{sf}(\vec{\alpha}, \Delta)$, then $\chi \in \mathrm{sf}(\vec{\alpha}^\frown j, \Delta)$.

Thus for each occurrence of a subformula χ there exists a unique sequence $\vec{\alpha}$ such that $\chi \in \mathrm{sf}(\vec{\alpha}, \Delta)$. (The reader should not be mislead by the notation 'χ' to think of a formula; rather, one should think of it as an occurrence of a subformula. Otherwise, $\vec{\alpha}$ is not unique.)

Theorem 4.7 (Hemaspaandra) *It can be checked in $O(||\Delta||)$ space whether a given set Δ of formulae is satisfiable in (polymodal) K.*

Now we shall deal with global tableaux. In a global tableau for $\Theta \dagger \Delta$, a node contains only members from $\mathrm{sf}(\Delta; \Theta)$. Let n be the cardinality of this set. As remarked earlier, we can design the calculus in such a way that no formula occurs twice in a sequence. Now, a formula may or may not occur in a set, and it may occur negated or unnegated. Hence there are at most 3^n different labels. If this is so, then we immediately see that if a branch is of length $\geq 3^n$ and closed, then there is a branch of length $< 3^n$ that is also closed. This means that a branch of length $\geq 3^n$ may simply be regarded as open. This bound can be sharpened. We let $p := \sharp\Theta$ and $q := \sharp\Delta$. Then branches need only be $3^p + q$ deep. This is immediately lowered to $2^p + q$ by noting that if a branch closes with Δ, it closes with any superset $\Delta' \supseteq \Delta$. In other words, partiality may help to keep branches short, but it does not increase the length. This is cognate to the following theorem, which is a refinement of Lemma 3.1.9 of [9] (for which almost literally the same proof can be used).

Theorem 4.8 *Let Δ be a set of formulae and φ a formula. Then put $p := \sharp\Delta$ and $q := \sharp\varphi$. Then $\Delta \Vdash_{\mathsf{K}} \varphi \quad \Leftrightarrow \quad \Box^{<2^p+q+1}\Delta \vdash_{\mathsf{K}} \varphi$.*

Corollary 4.9 *Let $q < \omega$ and $\mathsf{K} := \mathsf{K}_q$. It can be checked in $O(2^n \log n)$–space whether or not $\Delta \Vdash_{\mathsf{K}} \varphi$, where $n := \sharp(\Delta; \varphi)$.*

This follows from the fact that the length of branches has an exponential upper bound. Notice that this length bound cannot be significantly reduced.

In [10] we have shown that the number of ($\Box$E)–steps is in some cases $O(2^{c\sqrt{n}})$ for some $c > 0$, and this means that it is unlikely that one can do with less than exponential space. It can be shown on the other hand that there is an exponential time algorithm checking global satisfiability. Moreover, the following is also known (see [14]).

Theorem 4.10 (Spaan) *K is globally EXPTIME–complete.*

Notice that this result holds also if a different measure of length is taken, namely the number of subformulae. This is of some importance later on. From this we shall obtain upper bounds for the global time complexity for many modal logics.

5 Space Bounds via Reduction Functions

Lemma 5.1 *Let $L \supseteq \mathsf{K}$. Then the following holds with respect to the number of subformulae, with respect to the length, and the modified length. Let X be a global reduction function from L to K. If X is polynomial, L is globally EXPTIME.*

The global reduction functions defined above are quadratic in the (modified) length. Hence it is established that the logics discussed here are generally globally in EXPTIME. This follows from the following observation. Suppose that $L = \mathsf{K} \oplus \Xi$. It is easy to see that $X(\Delta)$ consists of substitution instances of some members of Ξ. Thus, all we need to know is what to substitute for the variables. The functions given earlier have the property that the formulae that are substituted for the variables are from $\mathrm{sf}^{\neg}(\Delta)$. For example, $X_{\mathsf{B}}(\Delta) = \{\neg\Box\chi \to \Box\neg\Box\chi : \Box\chi \in \mathrm{sf}(\Delta)\}$ is obtained by substituting $\neg\Box\chi$ for p in $p \to \Box\neg\Box\neg p$, and $\mathsf{B} = \mathsf{K} \oplus p \to \Box\Diamond p$. Of course, $\Diamond$ abbreviates $\neg\Box\neg$; and we have replaced $\neg\neg\Box\chi$ by $\Box\chi$, but this is harmless cosmetics. This motivates the following.

Definition 5.2 *Let X be a reduction function. Suppose that there is a finite set Θ such that $X(\Delta)$ consists of some or all substitution instances of Θ by members of $\mathrm{sf}^{\neg}(\Delta)$ for its variables. Further, let n be the number of variables in Θ. Then we call X an n–**analytic global reduction function** with **skeletal set** Θ.*

It is checked that all reduction functions defined above are 1–analytic.

Lemma 5.3 *Let X be n–analytic with skeletal set Θ. Then $\sharp X(\Delta) \leq \sharp\Theta \cdot (\sharp\Delta)^n$.*

Theorem 5.4 *Let L and M be modal logic. Suppose that X is an n–analytic skeletal global reduction function from M to L. Then if L is globally in EXPTIME with respect to any of the measures, then so is M.*

Clearly, this theorem can be generalized in the following way. If X is a global reduction function from L to M satisfying the analogous conditions, and if M is in $\mathcal{C}$, then so is L given that X is $\mathcal{C}$–computable. Notice that if X reduces L to M and Y reduces M to N, then $Y \circ X$ reduces L to N. Moreover, the property required in the previous theorem is also preserved. So, the reductions can be cascaded. This is needed when we want to reduce G to K, for example. The reduction functions given are only from G to $\mathsf{K4}$. However, we also have reduction functions from $\mathsf{K4}$ to K. We shall refrain in the sequel from noting these obvious generalizations.

The local satisfiability problem still needs discussion. It will be treated by factoring the local reduction functions into a global reduction function plus a function bounding the depth of the tableau. The reduction functions given above are 1–analytic. This is not always so, for example with $\mathsf{K4.3}$. The following is a global reduction function for $\mathsf{K4.3}$ to $\mathsf{K4}$:

$$\begin{aligned}(22)\quad X_3(\Delta) := \{\neg\Box\chi \wedge \neg\Box\chi' \to \neg\Box(\chi \vee \Box\chi') \vee \neg\Box(\chi' \vee \Box\chi)\\ \vee \neg\Box(\chi \vee \chi') : \Box\chi, \Box\chi' \in \mathrm{sf}(\Delta)\}\end{aligned}$$

Of course it requires proof that this is a reduction function. Notice however that these formulae are instances of the characteristic axiom: replace p by $\neg\chi$ and q by $\neg\chi'$ in

$$(23)\qquad \Diamond p \wedge \Diamond q \to \Diamond(p \wedge \Diamond q) \vee \Diamond(p \wedge \Diamond p) \vee \Diamond(p \wedge q)$$

X_3 is skeletal but only 2–analytic. (In fact, there does not exist a 1–analytic reduction function; otherwise $\mathsf{S4.3}$ would have interpolation, contrary to fact.)

Lemma 5.5 *Let X be a global reduction function from M to L. Then there exists a function ρ from sets of formulae to natural numbers such that*

$$(24)\qquad Y(\Delta) := \{\Box^{<\rho(\Delta)+1}\vartheta : \vartheta \in X(\bigwedge \Delta \to \varphi)\}$$

is a local reduction function from M to L.

Proof. $\Delta \vdash_M \varphi$ implies $\Vdash_M \bigwedge\Delta \to \varphi$, which in turn implies $X(\bigwedge\Delta \to \varphi) \Vdash_L \bigwedge\Delta \to \varphi$ and so $\Box^{<\mu}X(\Delta;\varphi) \vdash_L \bigwedge\Delta \to \varphi$ for some number μ depending only on $\bigwedge\Delta \to \varphi$. Put $\rho(\bigwedge\Delta \to \varphi) := \mu$. Then $\Delta; \Box^{<\mu}X(\bigwedge\Delta \to \varphi) \vdash_L \varphi$, as promised. ■

There is a slight problem in the definition of ρ. We have defined ρ such that $\rho(\bigwedge\Delta \to \varphi) := \mu$, while technically ρ should depend only on $\Delta; \varphi$ (that is, not knowing what is premiss and what is conclusion). This can be dealt with in two ways: (a) We make the reduction functions sensitive to this distinction (which gives more subtle bounds), (b) we take the maximum over all numbers for partitions of $\Delta; \varphi$ into a conclusion and a premiss set. (b) is less optimal but asymptotically the difference can usually (and in the present cases definitely) be ignored.

Definition 5.6 *Let M and L be modal logics, X, Y and p as in the previous lemma. If X is n–analytic with skeletal set Θ Y is also called n–**analytic** and **depth reduction function** ρ.*

Notice that the local reduction need not be skeletal even if X is. However, there is an important case in which Y is once again skeletal.

Definition 5.7 *A reduction function X **splits** if*

$$X(\Delta; \varphi \wedge \psi) = X(\Delta; \varphi \to \psi) = X(\Delta; \varphi; \psi) \tag{25}$$

The following is shown in [9] and establishes the relevance of this concept.

Theorem 5.8 *Suppose that X is a splitting global reduction function from M to L. Then if L has interpolation, so does M.*

If X is splitting, $X(\bigwedge\Delta \to \varphi) = X(\Delta; \varphi)$. All reduction functions defined above are in fact splitting and skeletal.

Proposition 5.9 *Suppose that X is a splitting n–analytic global reduction function with skeletal set Θ. Further assume that ρ is a function from sets of formulae to numbers such that*

$$Y(\Delta) := \{\Box^{<\rho(\Delta)+1}\vartheta : \vartheta \in X(\Delta)\} \tag{26}$$

is a local reduction function from M to L. Then Y is n–analytic with skeletal set Θ, and splitting. Moreover,

$$\begin{aligned} \sharp Y(\Delta) &\leq 2\rho(\Delta)\,\mathrm{card}(X(\Delta)) + \sharp X(\Delta) \\ &\leq 2\rho(\Delta)(\sharp\Delta)^n + (\sharp\Delta)^n \sharp\Theta \end{aligned} \tag{27}$$

Notice that for the logics discussed in this paper, $\rho(\Delta) \leq \sharp\Delta$. So, if $n > 0$, the second term wins for large $\sharp\Delta$. This is still suboptimal. Define

$$Y'(\Delta) := \Box^{<\rho(\Delta)+1} \bigwedge X(\Delta) \tag{28}$$

Then

(29) $$\sharp Y'(\Delta) \leq 2\rho(\Delta) + \sharp(\bigwedge X(\Delta)) \leq 2\rho(\Delta) + 2\sharp\Theta(\sharp\Delta)^n$$

In order to use Hemaspaandra's results we need to bound the number $\sharp_\delta(Y(\Delta))$. However, notice that for any set Γ of formulae $\sharp_\delta(\Gamma) \leq \sharp\Gamma \cdot \mathrm{dg}(\Gamma)$.

Theorem 5.10 *Let $n > 0$. Suppose that Y is an n–analytic local reduction function from M to K with skeletal set Θ and depth reduction function ρ. Assume that $\rho(\Delta) \leq \sharp\Delta$. Then for given Δ a tableau can be computed using $O((\sharp\Delta)^n \cdot \mathrm{dg}(\Delta))$ space.*

Proof. For checking satisfiability, we check whether $\Delta \vdash_L \bot$. We construct a tableau for $\Sigma := \Delta; Y'(\Delta; \bot)$ in place of Δ. $\sharp_\delta(\Sigma) \leq \mathrm{dg}(\Sigma) \cdot \sharp\Sigma$, which is asymptotically of the magnitude $O(\mathrm{dg}(\Delta)(\sharp\Delta)^n)$. This is also the space bound. ■

In certain cases we can eliminate the additional factor $\mathrm{dg}(\Delta)$. X_{D} for example is 0–analytic. Here, $Y(\Delta) := \Box^{<\mathrm{dg}(\Delta)+1}\neg\Box\bot$, which has length $O(\mathrm{dg}(\Delta))$. Hence the reduction function adds sublinear material.

Corollary 5.11 *Satisfiability of Δ in $\mathsf{K.D}$ is in $O(||\Delta||)$ space.*

This can be sharpened to $O(\sharp_\delta\Delta)$. For other logics Hemaspaandra's method gives $O(\sharp\Delta \cdot \mathrm{dg}(\Delta))$. A second case is reduction functions which are degree homogeneous, such as X_{alt_1}. Here the following is a local reduction function:

(30) $$Y_{\mathsf{alt}_1}(\Delta) := \{\Box^d(\neg\Box\chi \to \Box\neg\chi) : \Box\chi \in \mathrm{sf}(d, \Delta), d \leq \mathrm{dg}(\Delta)\}$$

Also here we can apply a little bit of cosmetics. Put

(31) $$\begin{aligned} Y^*_{\mathsf{alt}_1}(\Delta) := \quad & \bigwedge X_{\mathsf{alt}_1}(\mathrm{sf}(0,\Delta)) \\ & \wedge (\Box \bigwedge X_{\mathsf{alt}_1}(\mathrm{sf}(1,\Delta)) \\ & \wedge (\Box \bigwedge X_{\mathsf{alt}_1}(\mathrm{sf}(2,\Delta)) \\ & \wedge \ldots)) \end{aligned}$$

Here, every member of $\mathrm{sf}(d, \Delta)$ is multiplied by two occurrences, and both have the same depth of embedding. In addition, there are $O(\sharp\Delta)$ new occurrences of subformulae. Thus, $\sharp_\delta Y^*_{\mathsf{alt}_1}(\Delta)$ is linear in $\sharp_\delta\Delta$. Thus we have shown the following result (which can also easily be shown by adapting Hemaspaandra's calculus).

Theorem 5.12 $\mathsf{K.alt}_1$ *is locally in* $O(||\Delta||)$*–space.*

Corollary 5.13 *Let* M *be any union of the following logics:* K, $\mathsf{K.T}$, $\mathsf{K.B}$, $\mathsf{K.alt}_1$, $\mathsf{K.D}$*. Then* M *is locally in* $O(\sharp\Delta\,\mathrm{dg}(\Delta))$*–space.*

For a proof we only need to show that the depth reduction function is linear in $\sharp\Delta$. This in turn means that the length of a maximal path in a minimal model for a consistent formula set must be linear in $\sharp\Delta$. For the model is basically created in all these cases from bundling open paths in open K–tableaux into a model (and defining the relation suitably). Clearly, for the postulates T, B, D, and alt_1 this is satisfied. A model needs to be only as deep as the modal depth of the formula set. (See the model construction procedure of [9].) By way of example, we give a proof of the fact that X_B is a skeletal reduction function with linear depth reduction function.

Theorem 5.14 *Let* L *be a subframe logic whose class of frames is closed under passing from the relation to its symmetric closure. Then* X_B *is a 1–analytic skeletal global reduction function of* $L.\mathsf{B}$ *to* L*. Moreover,* $Y_\mathsf{B}(\Delta) := \Box^{<\mathrm{dg}(\Delta)+1}X_\mathsf{B}(\Delta)$ *is a local reduction function.*

Proof. The skeletality of X_B is easy to see. We prove that Y_B is a local reduction function; it follows directly that X_B is a global reduction function. Assume that $\Delta; \Box^{\leq q}X_\mathsf{B}(\Delta;\varphi)$ is L–consistent, where $q := \sharp\Delta$. We will show that Δ is $L.\mathsf{B}$–consistent, the converse being easy. Pick an L–frame $\mathfrak{F} = \langle F, \lhd\rangle$, a valuation β and a world x such that $\langle\mathfrak{F},\beta,x\rangle \vDash \Delta; \Box^{\leq q}X_\mathsf{B}(\Delta;\varphi)$.

Let $\mathfrak{F}^B$ be obtained by replacing $\lhd$ by its symmetric closure, $\blacktriangleleft$. $\mathfrak{F}^B \vDash L.\mathsf{B}$, by assumption on L. By induction on χ we show that for all w reachable in at most $q - \mathrm{dg}(\chi)$ steps from x:

$$\langle\mathfrak{F}^B,\beta,w\rangle \vDash \chi \qquad \Leftrightarrow \qquad \langle\mathfrak{F},\beta,w\rangle \vDash \chi \tag{32}$$

The only critical step is $\chi = \Box\tau$. From left to right this follows from the fact that if $x \lhd y$ then also $x \blacktriangleleft y$. For the other direction, assume we have $\langle\mathfrak{F}^B,\beta,w\rangle \nvDash \Box\tau$. Then there is a v such that $w \blacktriangleleft v$ and $\langle\mathfrak{F}^B,\beta,v\rangle \vDash \neg\tau$. If $w \lhd v$, we are done. Otherwise, $v \lhd w$. However, we have $\langle\mathfrak{F},\beta,v\rangle \vDash \neg\tau \to \Box\neg\Box\neg\tau$. (For $x \vDash \Box^{\leq q}(\neg\tau \to \Box\neg\Box\neg\tau)$, and v is reachable in at most q steps from x.) So, if $\langle\mathfrak{F},\beta,v\rangle \vDash \neg\tau$, we have $\langle\mathfrak{F},\beta,w\rangle \vDash \neg\Box\tau$. This means that $\langle\mathfrak{F},\beta,w\rangle \nvDash \Box\tau$, as promised.

Now let G be the set of all points reachable from x in at most q steps; let $\mathfrak{G}$ be the induced subframe of $\mathfrak{F}^B$ and let $\gamma(p) := \beta(p) \cap G$. Since L is a subframe logic, $\mathfrak{G}$ is a frame for L. It is symmetric, therefore a frame for $L.\mathsf{B}$. We claim $\langle\mathfrak{G},\gamma,x\rangle \vDash \Delta$. Namely, the following is established by an easy induction: for all points y reachable in at most p steps from x and all subformulae χ of Δ of depth $\leq q-p$, $\langle \mathfrak{G}, \gamma, y\rangle \vDash \chi$ iff $\langle \mathfrak{F}^B, \gamma, y\rangle \vDash \chi$. The claim now follows since Δ has degree q and therefore every member of it is true at w. Hence Δ is $L.\mathsf{B}$–consistent. ■

We remark that in the definition of Y_{B} we could have dropped all formulae of degree $> \mathrm{dg}(\Delta)$. This gives an improvement by a factor 1/2, which is however ignored in the O–notation.

6 Transitive Logics

A logic L is **transitive** if it contains the axiom $\mathsf{4}$: $\Box p \to \Box\Box p$. It is immediate that for transitive logics $\Delta \Vdash_L \varphi$ iff $\Delta; \Box\Delta \vdash_L \varphi$. This transformation of problems is linear. Hence, a transitive logic is globally $\mathcal{C}$–hard (globally in $\mathcal{C}$, globally $\mathcal{C}$–complete) if and only if it is locally $\mathcal{C}$–hard (locally in $\mathcal{C}$, locally $\mathcal{C}$–complete). Also, if X is a global reduction function from M to $L \supseteq \mathsf{K4}$, $Y(\Delta) := \Box^{<2}X(\Delta)$ is a local reduction function from M to L. One immediate consequence is the following.

Corollary 6.1 *Let $\mathcal{C}$ be closed under linear transforms. Then if* $\mathsf{K4}$ *is locally in $\mathcal{C}$, so is* G, $\mathsf{S4}$, $\mathsf{K4.D}$, *and* Grz.

Thus, we only need to establish the local complexity of $\mathsf{K4}$. We can try Hemaspaandra's methods again. In place of the rule $(\Box\mathrm{H})$ we take the following rule:

$$(33) \qquad (\Box\mathrm{H4}) \quad \frac{\Gamma; \Box\Delta; \neg\Box\Sigma; \neg\Box\chi}{(\Delta; \Box\Delta; \neg\chi)^*}$$

where Γ is a set of formulae of degree 0. Once again the calculus steps from downward saturated sets to downward saturated sets. A downward saturated set takes $O(\sharp\Delta)$ space to code. The formula on which the rule operates takes $O(\log ||\Delta||)$ space. Now observe the following. (a) If $\Box\delta$ appears above the line, it also appears below. (b) If $\neg\Box\delta$ appears below the line, it also appears above. So, the set below the line is characterized uniquely by the set of $\Box\delta$ that occur in it in addition to those that are above the line. Further, we we shall show it is not necessary to use the rule twice on the same formula $\neg\Box\chi$. Thus, for backtracking, we need a record only of (a) the variables occurring in the set, (b) the formulae $\Box\delta$ that are being added, (c) the formulae $\neg\Box\delta$ that are being retracted, (d) the formula $\neg\Box\chi$. So the tableau is stored as a sequence $\langle\langle A_i, B_i, C_i, \delta_i\rangle : i < n\rangle$, where A_i is a set of variables, B_i a set of subformulae of the form $\Box\delta$, C_i a set of subformulae or their negations of the form $\neg\Box\delta$, and δ_i a subformula. Then a variable is true at node i iff it is in A_i; a subformula $\Box\delta$ is true at

i iff it is in some B_j, $j < i$; finally, $\neg\Box\delta$ is true iff it is not in any C_j for $j < i$, and not equal to any of the δ_j, $j < i$. Clearly, since the B_i and the C_i are pairwise disjoint but otherwise of any size, they can only be globally estimated. The bound is $\sharp\Delta \cdot \log \sharp\Delta$. Given that the length of a branch is bounded by $\sharp\Delta$, the overall bound is $\sharp\Delta \cdot \mathrm{card}(\mathrm{var}(\Delta)) + 3\sharp\Delta \log \sharp\Delta$. Finally, look at the transition

$$\frac{\Gamma; \Box\Delta; \neg\Box\Sigma; \neg\Box\chi}{(\Delta; \Box\Delta; \neg\chi)^*} \tag{34}$$

The set Γ of nonmodal formulae does not influence the formula set below the line. So, A_j, $j < i$, is not needed in backtracking. In other words, we do not backtrack to another choice of the values for the variables. This means that all we need to do is to keep a record of $\langle\langle B_j, C_j, \delta_j\rangle : j < i\rangle$, which is of size $O(\sharp\Delta \log \sharp\Delta)$.

Theorem 6.2 (Nguyen) *Satisfiability in K4 can be checked using only* $O(\sharp\Delta \cdot \log \sharp\Delta)$ *space.*

Corollary 6.3 *Satisfiability in G, S4, Grz is in* $O(\sharp\Delta \cdot \log \sharp\Delta)$*–space.*

There is also a proof using global tableaux. This proceeds by bounding the number of applications of the rule ($\Box$E). We shall start with a global K–tableau for

$$X_4(\Delta); \{p \vee \neg p : p \in \mathrm{var}(\Delta)\} \dagger \Delta \tag{35}$$

Suppose that no closed tableau exists. (This is the same as to say that Δ does not globally follow from $X_4(\Delta)$ and $\{p \vee \neg p : p \in \mathrm{var}(\Delta)\}$.) Then we construct a model as follows. The nodes are all downward saturated sets occurring in any open branch of a tableau that has the form $\Sigma \dagger \Theta$ such that Σ did not appear in a previous node of that branch. By the fact that we have a tableau in which $p \vee \neg p$ is contained everywhere in Θ for each variable occurring in Δ, downward saturated sets will either contain p or $\neg p$ for these variables.

We put $\Theta \dagger \Sigma_1 \lhd \Theta \dagger \Sigma_2$ iff $\Theta \dagger \Sigma_1$ appeared above $\Theta \dagger \Sigma_2$ in that branch. Then we put $\blacktriangleleft := \lhd^+$, the transitive closure. This defines a frame. We define $\beta(p) := \{\Theta \dagger \Sigma : p \in \Sigma\}$. It can be checked that in this model, χ holds at $\Theta \dagger \Sigma$ iff it is in Σ. The model is transitive, and therefore we have a K4–model for Δ.

Lemma 6.4 *Suppose that b is a branch in a global K–tableau for*

$$X_4(\Delta); \{p \vee \neg p : p \in \mathrm{var}(\Delta)\} \dagger \Delta \tag{36}$$

Suppose further that (□E) *has least priority and that there exist no two downward saturated nodes* $\Theta \dagger \Sigma_1$ *and* $\Theta \dagger \Sigma_2$ *such that* $\Sigma_1 = \Sigma_2$. *Then* (□E) *has been applied at most* $3 \cdot 2^n \sharp\Delta$ *times, where* n *is the number of variables occurring in* Δ.

Proof. In a local tableau, the number of applications of (□E) within a branch is bounded by the modal depth of the formula. More precisely, it is bounded by the number of nested negative occurrences of □. In a global tableau, the left hand set feeds the right hand side each time (□E) is executed. However, the set that is being added is always the same. We have

$$X_4(\Delta) = \{\neg(\Box\chi \wedge \neg\Box\Box\chi) : \Box\chi \in \mathrm{sf}(\Delta)\} \tag{37}$$

A downward saturated set is of the form $O; P; N$, where $O \subseteq \{p, \neg p : p \in \mathrm{var}(\Delta)\}$, $P \subseteq \{\Box\chi, \Box\Box\chi : \Box\chi \in \mathrm{sf}(\Delta)\}$ and $N \subseteq \{\neg\Box\chi : \Box\chi \in \mathrm{sf}(\Delta)\}$. Suppose that some set contains $\Box\chi$ for some $\Box\chi \in \mathrm{sf}(\Delta)$. Then, since it is the saturation of a set containing $\neg(\Box\chi \wedge \neg\Box\Box\chi)$, it also contains $\Box\Box\chi$. Thus, after one application of (□E) we have that $\Box\chi$ is contained in the set below the line as well, and after saturation also $\Box\Box\chi$. Hence, the set P cannot shrink, while the set N cannot grow. So, in passing from $O; P; N$ to $O'; P'; N'$ we must have either $N \supsetneq N'$, or $P \subsetneq P'$, or $O' \neq O$. Moreover, as long as $N = N'$ and $P = P'$, O may never occur again. This yields the bound. ■

The bound obtained in this way is too rough for our purposes. However, notice that downward saturated sets correspond to nodes in a model. Moreover, if $N' = N$ and $P' = P$, this means that we move inside a cluster. The cluster size is at most 2^n. However, it is not necessary to form big clusters all the time. We can avoid forming nontrivial clusters in most cases. In effect, one can show that if there exists a closed tableau then there exists a closed tableau with the following property. If at some step $N = N'$ and $P = P'$, then there is a formula $\neg\Box\chi$ for which (□E) is applied, and this formula will never appear in a successor cluster again. In terms of models this means that if there exists a model then there exists a model such that if we have a nontrivial cluster where $\neg\Box\chi$ is true, then this formula will be false in all clusters succeeding that cluster. This is what we shall show now.

Let $\mathfrak{F} = \langle F, \lhd \rangle$ be a frame. A **path of length** n is a sequence $\Pi = \langle w_i : i < n + 1 \rangle$ such that $w_i \lhd w_{i+1}$ for all $i < n$. Π is **nonrepeating** if for no $i < j$, $w_i = w_j$. We shall show that the size of nonrepeating paths in a minimal model is linear in $\sharp\Delta$. The following proof is an adaptation of the proof methods in [10].

Lemma 6.5 *Let $M \supseteq \mathsf{K4}$ be a cofinal subframe logic. Then a consistent formula set Δ has a model in which every nonrepeating path has length $\leq \sharp\Delta$.*

Proof. Take a model $\langle \mathfrak{F}, \beta, x\rangle \vDash \Delta$. Let $\psi \in \mathrm{sf}(\Delta)$. Call a point y χ–**critical**, if (a) $\langle \mathfrak{F}, \beta, y\rangle \vDash \chi$ and (b) from $\langle \mathfrak{F}, \beta, z\rangle \vDash \chi$ and $y \lhd z$ follows $z \lhd y$. Evidently, if y and y' are χ–critical and $y \lhd y'$ then $y' \lhd y$ as well. Moreover, if y satisfies $\Diamond\chi$, then there is a critical z such that $y \lhd z$ and z satisfies χ. Take a cluster $C(u) = \{v : u \lhd v \lhd u\}$. If $|C(u)| = 1$, we remove $C(u)$ if it is not critical. If $|C(u)| > 1$, and it contains critical points, then we keep for each $\chi \in \mathrm{sf}(\Delta)$ exactly one χ–critical point, if the cluster contains one. If $|C(u)| > 1$ and $C(u)$ contains no critical points, we remove all points. Let G be the set of remaining points. Call the induced subframe $\mathfrak{G}$. Denote by γ the valuation induced by β on $\mathfrak{G}$. We claim that $\langle \mathfrak{G}, \gamma, x\rangle \vDash \Delta$. In fact, we show that for each formula $\chi \in \mathrm{sf}(\Delta)$ and each $y \in G$:

$$(38) \qquad \langle \mathfrak{G}, \gamma, y\rangle \vDash \chi \quad \Leftrightarrow \quad \langle \mathfrak{F}, \beta, y\rangle \vDash \chi$$

Indeed, if χ is a variable, this holds by definition of γ. The only nonobvious step is $\chi = \Diamond\chi'$. Assume that $\langle \mathfrak{G}, \gamma, y\rangle \vDash \Diamond\chi'$. Then there exists a $z \in G$ such that $y \lhd z$ and $\langle \mathfrak{G}, \gamma, z\rangle \vDash \chi'$. By induction hypothesis, $\langle \mathfrak{F}, \beta, z\rangle \vDash \chi'$ and so $\langle \mathfrak{F}, \beta, y\rangle \vDash \Diamond\chi' = \chi$. Now assume that the latter holds. Then there is a χ'–critical successor z of y. Therefore, $\langle \mathfrak{F}, \beta, z\rangle \vDash \chi'$ and since $z \in G$, we have $\langle \mathfrak{G}, \gamma, z\rangle \vDash \chi'$ by inductive hypothesis. Therefore $\langle \mathfrak{G}, \gamma, y\rangle \vDash \Diamond\chi' = \chi$.

It is clear that a point is χ–critical in $\langle \mathfrak{F}, \beta\rangle$ iff it is χ–critical in $\langle \mathfrak{G}, \gamma\rangle$. Finally, we count the number of points in a nonrepeating path. Let $y, y' \in G$, and $y \lhd y'$. Since y and y' are critical they must be critical for different formulae. Hence we have at most $\sharp\Delta$ points in a nonrepeating path. ■

This gives the following.

Theorem 6.6

$$(39) \qquad \Delta \vdash_{\mathsf{K4}} \varphi \quad \Leftrightarrow \quad \Delta; \Box^{<\sharp(\Delta;\varphi)+1} X_4(\Delta;\varphi) \vdash_{\mathsf{K}} \varphi$$

Using Theorem 5.10 we get

Corollary 6.7 *K4 is in $O(\sharp\Delta\, \mathrm{dg}(\Delta))$–space.*

It follows with the help of the previous results that also G, Grz, and $\mathsf{S4}$ are in $O(\sharp\Delta\, \mathrm{dg}(\Delta))$–space. This is somewhat different from the bound $O(||\Delta|| \log ||\Delta||)$.

Using the formulae of [10] polynomial space bounds can be obtained for all subframe logics, the degree of the polynomial equal to the number of variables needed to axiomatize that logic. This is not entirely straightforward as they result from substituting into the skeletal set some complex formulae built essentially from maximal consistent subsets $\mathrm{sf}^{\neg}(\Delta)$. However, with a set Δ given, a subset is linearly codable, and so the substitution instances need only $O((\sharp\Delta)^n)$–space to code.

It is known that extensions of $\mathsf{S4.3}$ are cofinal subframe logics (see [2]). For an extension of $\mathsf{S4.3}$ only a single tableau needs to be computed. Hence, such an extension is in NP and therefore NP–complete if consistent (see [14]). The space bounds established here are $O((\sharp\Delta)^k \log \sharp\Delta)$, where k is the number of variables needed to axiomatize the logic. Hence $k = 2$ for $\mathsf{S4.3}$ itself. This may in many cases be suboptimal.

In a similar fashion, a lot more results can be shown. We can prove that satisfiability in tense logic is in $O(||\Delta|| \log \sharp\Delta)$, and that PDL with converse is EXPTIME–complete (see [4]). Furthermore, the technique of [8] shows that many splitting axioms preserve complexity bounds above $\mathsf{K4}$ and $\mathsf{S4}$. This applies in particular (above $\mathsf{K4}$) to .1, .2 and (above $\mathsf{S4}$) $.\mathsf{Dum}$, to name a few. Using the standard Gödel–translation, which is linear, we get from Corollary 6.3 the space bound $O(||\Delta|| \log ||\Delta||)$ not only for intuitionistic propositional logic, obtained in [6], but also for extensions of Int (for example, $\mathsf{KC} = \mathsf{Int} + \neg p \vee \neg\neg p$, the logic corresponding to $\mathsf{S4.2}$, or $\mathsf{LC} = \mathsf{Int} + p \to q \vee q \to p$, the logic corresponding to $\mathsf{S4.3}$).

7 Conclusion

Although there exist quite sophisticated tableau calculi specially adapted for logics extending K, the known (worst case) complexity bounds can typically be established using a purely combinatorial method based on reduction functions. The method may not be as useful in actual applications; however, it has the advantage of being uniform, and it can be generalized in various ways. A straightforward application of the method typically yields only $O(\sharp\Delta \cdot \mathrm{dg}(\Delta))$, but this latter bound is not necessarily worse than the standardly proved $O(||\Delta|| \log ||\Delta||)$ (which typically can be improved to $O(\sharp\Delta \log \sharp\Delta)$ at no cost). Also, based on tableau methods, the latter can typically easily be established. Moreover, in the transitive case we obtain the latter bound directly as well, using a modification of Hemaspaandra's calculus adapted for $\mathsf{K4}$.

BIBLIOGRAPHY

[1] D. Basin, S. Matthews, and L. Viganò. A new method for bounding the complexity of modal logic. In G. Gottlob, A. Leitsch, and D. Mundici, editors, *Proceedings of the 5th Kurt Gödel Colloquium on Computational Logic and Proof Theory (KGC'97),*

number 1289 in Lecture Notes in Computer Science, pages 89 – 102, Heidelberg, 1997. Springer.

[2] Alexander Chagrov and Michael Zakharyaschev. *Modal Logic*. Oxford University Press, Oxford, 1997.

[3] Melvin Fitting. *Proof Methods for Modal and Intuitionistic Logic*. Number 169 in Synthese Library. Reidel, Dordrecht, 1983.

[4] Giuseppe de Giacomo. Eliminating "Converse" from Converse PDL. *Journal of Logic, Language and Information*, 5:193 – 208, 1996.

[5] Edith Hemaspandra. Modal Satisfiability is in Deterministic Linear Space. In *Computer Science Logic*, number 1786 in Lecture Notes in Computer Science, pages 332 – 343, Heidelberg, 2000. Springer Verlag.

[6] Jörg Hudelmaier. An $O(n \log n)$–Space Decision Procedure for Intuitionistic Propositional Logic. *Journal of Logic and Computation*, 3:63 – 75, 1993.

[7] Jörg Hudelmaier. Improved Decision Procedures for the Modal Logics K, T and S4. In H. Kleine Büning, editor, *Proceedings of CSL '95*, number 1092 in Lecture Notes in Computer Science, pages 320 – 334, 1996.

[8] Marcus Kracht. Splittings and the finite model property. *Journal of Symbolic Logic*, 58:139 – 157, 1993.

[9] Marcus Kracht. *Tools and Techniques in Modal Logic*. Number 142 in Studies in Logic. Elsevier, Amsterdam, 1999.

[10] Marcus Kracht. Reducing Modal Consequence Relations. *Journal of Logic and Computation*, 11:879 – 907, 2001.

[11] R. E. Ladner. The computational complexity of provability in systems of modal logic. *SIAM Journal of Computing*, 6:467 – 480, 1977.

[12] Linh Anh Nguyen. A New Space Bound for the Logics K4, KD4, and S4. In M. Kutyłowski, L. Pacholski, and T. Wierzbicky, editors, *Proceedings of MFCS'99*, LNCS 1675, pages 321 – 331, 1999.

[13] Wolfgang Rautenberg. Modal tableau calculi and interpolation. *Journal of Philosophical Logic*, 12:403 – 423, 1983.

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

[15] L. Viganò. *A framework for non–classical logics*. PhD thesis, Universität des Saarlandes, Saarbrücken, Germany, 1997.