Terminating tableau calculi for modal logic K with global counting operators

Mohammad Khodadadi, Renate A. Schmidt, Dmitry Tishkovsky

School of Computer Science, The University of Manchester, United Kingdom

Michał Zawidzki

Department of Logic, University of Lodz, Poland
School of Computer Science, The University of Manchester, United Kingdom

Abstract

This paper presents the first systematic treatment of tableau calculi for modal logic K with global counting operators. Using a recently introduced tableau synthesis framework we establish two terminating tableau calculi for the logic. Whereas the first calculus is a prefix tableau calculus, the second is a refinement that internalises the semantics of the logic without using nominals. We prove the finite model property for the logic and show that adding the unrestricted blocking mechanism does not break soundness and completeness of the calculi and ensures termination in both cases. We have successfully implemented the prefix tableau calculus in the MetTeL2 tableau prover generation platform.

Keywords: modal logic, hybrid logic, tableau, counting operators, finite model property.

1 Introduction

Counting modalities were first introduced by Fine in [8] under the name of *graded modalities*. They allowed expressing a number of successors of a particular world, at which a certain formula holds. In particular, a formula $\Diamond_{=n}\top$ expresses the fact that the current world has exactly n successors. Some further developments of the theory of graded modalities can be found in [4,5,7]. Van der Hoek and de Rijke [18] established graded modal logics as a modal tool for investigating first-order counting quantifiers and introduced the notion of *propositional logic with counting* ($\mathcal{PLC}$) as a name for the logic S5 with graded modalities (see also [3]).

With the aim of improving the expressivity of modal logics Areces et al. [2] introduced modal logics with counting operators ($\mathcal{MLC}$). Global counting operators $\mathsf{E}_{>n}$, $\mathsf{E}_{<n}$ and $\mathsf{E}_{=n}$ were added to a modal language with the ordinary modalities. Global counting operators increase the expressive power of a logic by allowing nominals, the universal modality, and counting the cardinality of

a domain (by a formula $\mathsf{E}_{=n}\top$). It also enables the formalisation of natural language queries that involve numbers.

In this paper we provide tableau-based decision procedure for modal logic K with global counting operators, referred to as $\mathrm{K}(\mathsf{E})_n$. Having all the properties mentioned in the forgoing paragraph, $\mathrm{K}(\mathsf{E})_n$ is a powerful extension of the ordinary modal logic K. Introducing counting operators not only allows encoding various interesting problems within the language of the logic (e.g., finite tiling problems) but covers also the expressive power of many modal logic (graded modal logics, most hybrid logics). However, a detailed study of the expressivity of $\mathrm{K}(\mathsf{E})_n$ is beyond the scope of this paper.

In the existing literature several approaches for deciding modal/description logics with counting operators can be found. [2] describes a decision procedure for modal logics with counting operators that exploits the translation function from the modal counting language to the hybrid language with the universal modality $\mathcal{H}(\mathsf{A})$. In [19] a sound and complete axiomatisation for $\mathrm{K}(\mathsf{E})_n$ is provided, which gives a basis for a standard Hilbert-style calculus. More direct, tableau-based decision procedures for $\mathcal{MLC}$ were established in the field of description logics where counting operators are known under the guise of *cardinality constraints*. Sound, complete and terminating tableau-calculi for these logics can be found in [10,11,6]. These calculi, in general, do not differ in the rules for cardinalities, however, they utilise different blocking mechanisms for ensuring termination, such as pairwise blocking or pattern-based blocking.

We exploit the framework from [15] to synthesise a sound, complete and terminating prefix tableau calculus for $\mathrm{K}(\mathsf{E})_n$. We also provide a refinement of this calculus consisting in *internalising* the semantics of the logic within the language of the logic. This is the first calculus that deals with global counting operators in purely modal terms. Even though in [11] an internalised tableau calculus for a logic with cardinality constraints is presented, the tableau rules use syntactic entities of a separate sort, namely *nominals*, to encode the semantics within the language of the logic, whereas we only exploit global counting operators to dispense with meta-linguistic expressions like $\mathfrak{M}, x \models \varphi$ or $R(x, y)$. Termination for our calculus is obtained by using the *unrestricted blocking* rule (ub) and by the fact that the logic $\mathrm{K}(\mathsf{E})_n$ has the *finite model property*, which is proven in this paper. We show that although the (ub)-rule is generic, our calculus still remains complexity-optimal.

We also describe a successful implementation of the calculus using the MetTeL2 tableau prover generator [17,1].

The paper is structured according to the steps in the tableau synthesis approach (see [14] for an overview of the framework). The first step is the specification of the syntax and semantics of the logic $\mathrm{K}(\mathsf{E})_n$. This involves defining an object language and a many-sorted first-order meta-language for $\mathrm{K}(\mathsf{E})_n$. This is done in Section 2. Section 3 describes the next step, which is the synthesis of a tableau calculus from the specification of $\mathrm{K}(\mathsf{E})_n$. Sections 4 and 5 perform the checks needed to establish soundness and completeness of the calculus in the framework. To obtain termination a proof of the finite model property for $\mathrm{K}(\mathsf{E})_n$ is given. In Sections 6 and 7 we focus on possible refinements of the calculus, one of which is obtained by internalising the semantics. Section 8 briefly describes a MetTeL2-implementation of the prefix tableau calculus. In Section 9 we discuss related work on tableau approaches for reasoning with counting operators. Conclusions and prospective work are presented in Section 10.

2 The logic $\mathbf{K}(\mathsf{E})_n$

First, we need to define the *object language* of $\mathrm{K}(\mathsf{E})_n$. Let $\textsc{prop} = \{p_1, p_2, \ldots\}$ be a countable set of propositional letters. We define a set $\textsc{form}$ of formulas of $\mathrm{K}(\mathsf{E})_n$ as follows:

$$\textsc{form} ::= \; p \;\mid\; \neg\varphi \;\mid\; \varphi \wedge \psi \;\mid\; \Diamond\varphi \;\mid\; \mathsf{E}_{>n}\varphi, \qquad (\mathrm{K}(\mathsf{E})_n)$$

where $p \in \textsc{prop}$, $\varphi \in \textsc{form}$, $n \in \mathbb{N}$. We also give explicit definitions of other Boolean connectives and modal operators, since they can serve as rewrite rules in the tableau calculi.

$$\varphi \vee \psi := \neg(\neg\varphi \wedge \neg\psi) \qquad \varphi \to \psi := \neg(\varphi \wedge \neg\psi) \qquad \Box\varphi := \neg\Diamond\neg\varphi$$

$$\mathsf{E}_{<n}\varphi := \neg\mathsf{E}_{>n-1}\varphi \qquad \mathsf{E}_{=n}\varphi := \mathsf{E}_{>n-1}\varphi \wedge \neg\mathsf{E}_{>n}\varphi$$

A *model* for $\mathrm{K}(\mathsf{E})_n$ is a triple $\langle W, R, V\rangle$ where W is a non-empty set, R is a binary relation on W, $V : \textsc{prop} \to \mathcal{P}(W)$ is a valuation function assigning to each $p \in \textsc{prop}$ a set of worlds $w \in W$ in which p holds. Given a model $\langle W, R, V\rangle$ and $w \in W$, the semantics for $\mathbf{K}(\mathsf{E}_n)$ is defined as follows:

$$\begin{array}{lll}
\mathfrak{M}, w \models p & \text{iff} & w \in V(p),\; p \in \textsc{prop} \\
\mathfrak{M}, w \models \neg\varphi & \text{iff} & \mathfrak{M}, w \not\models \varphi \\
\mathfrak{M}, w \models \varphi \wedge \psi & \text{iff} & \mathfrak{M}, w \models \varphi \text{ and } \mathfrak{M}, w \models \psi \\
\mathfrak{M}, w \models \Diamond\varphi & \text{iff} & \text{there is a } v \text{ such that } wRv \text{ and } \mathfrak{M}, v \models \varphi \\
\mathfrak{M}, w \models \mathsf{E}_{>n}\varphi & \text{iff} & \mathsf{Card}(\{w \mid \mathfrak{M}, w \models \varphi\}) > n,
\end{array} \qquad (1)$$

$\mathsf{Card}(A)$ denotes the cardinality of the set A.

The *meta-language* of the tableau synthesis framework to specify the semantics of $\mathrm{K}(\mathsf{E})_n$ is a many-sorted first-order language. Following [15] henceforth we denote it by $\mathcal{FO}(\mathrm{K}(\mathsf{E})_n)$. A word of explanation needs to be devoted to the notion of *sort*. Sorts of a particular language divide expressions of the language into distinct sets. In the case of modal logics, we can identify the sort of a particular expression φ with the arity of first-order predicates resulting from the standard translation applied to φ. This means that nominals would have sort 0 as they translate to constants (nullary predicates), propositional variables would have sort 1 (as they become unary predicates after translation). Roles in description logics or programs in PDL would have sort 2 (since their first-order logic equivalents are binary relations). For the logic $\mathrm{K}(\mathsf{E})_n$ we need only the sort 1 for formulas (as neither nominals nor role expressions are

$S^b: x \approx x$	$x \approx y \rightarrow y \approx x$	$x \approx y \wedge y \approx z \rightarrow x \approx z$
$(x \approx y \wedge \nu(p,x)) \rightarrow \nu(p,y)$		$(x \approx y \wedge R(x,z)) \rightarrow R(y,z)$
$(x \approx y \wedge R(z,x)) \rightarrow R(z,y)$		$x \approx y \rightarrow f(p,x) \approx f(p,y)$

Figure 1: Semantic specification of background theory S^b for $\mathrm{K}(\mathsf{E})_n$

$$
\begin{aligned}
S^0: \; & \forall x(\nu(\neg\varphi, x) \equiv \neg\nu(\varphi, x)) \\
& \forall x(\nu(\varphi \wedge \psi, x) \equiv \nu(\varphi, x) \wedge \nu(\psi, x)) \\
& \forall x(\nu(\Diamond\varphi, x) \equiv \exists z(R(x,y) \wedge \nu(\varphi, y))) \\
& \forall x(\nu(\mathsf{E}_{>n}\varphi, x) \equiv \exists y_1 \ldots \exists y_{n+1}(\textstyle\bigwedge_{0<i\leq n+1} \nu(\varphi, y_i) \wedge \bigwedge_{0<i<j\leq n+1}(y_i \not\approx y_j)))
\end{aligned}
$$

Figure 2: Semantic specification of connectives S^0 for $\mathrm{K}(\mathsf{E})_n$

present in the language). The sorts of the meta-language $\mathcal{FO}(\mathrm{K}(\mathsf{E})_n)$ are then the sort 1 for formulas of $\mathrm{K}(\mathsf{E})_n$ and the sort 2 as the domain sort. Expressions of sort 2 relate to the elements of the first-order domain (under a particular valuation). $\mathcal{FO}(\mathrm{K}(\mathsf{E})_n)$ as a first-order language is equipped with the usual logical constants and expressions. Furthermore, it contains the following symbols as first-order equivalents for $\mathrm{K}(\mathsf{E})_n$-expressions:

(i) functional symbols obtained from the connectives of $\mathrm{K}(\mathsf{E})_n$: unary functional symbols $\neg$, $\Diamond$, $\mathsf{E}_{>n}$ (sort $1 \longrightarrow 1$), and binary functional symbol $\vee$ (sort $1, 1 \longrightarrow 1$);
(ii) a constant binary predicate symbol R (sort $(2,2)$);
(iii) the equality symbol $\approx$ (sort $(2,2)$);
(iv) an interpretation predicate symbol ν (sort $(1,2)$).

In the semantic specification $\mathrm{K}(\mathsf{E})_n$-formulas (of sort 1) are treated as $\mathcal{FO}(\mathrm{K}(\mathsf{E})_n)$-terms. The ν predicate can be viewed as representing the $\models$ relation, this means the formula $\nu(\varphi, x)$ expresses that $\mathfrak{M}, x \models \varphi$.

The foregoing instruments are sufficient to give a full *semantic specification of* $\mathrm{K}(\mathsf{E})_n$ *in* $\mathcal{FO}(\mathrm{K}(\mathsf{E})_n)$. We can look upon the semantic specification of a certain logic as a set of axioms for a class of first-order structures where each connective from the logic is unambiguously defined (see [15]). It consists of two parts: the background theory (S^b) and the definitions of the connectives (S^0). The background theory provides a frame characterisation for the considered logic. In our case it only contains equality axioms, since for us the base logic is modal logic K for arbitrary frames. The background theory conditions are given in the usual manner, in a universally quantified form. Obtaining tableau rules for the background theory involves eliminating quantifiers using Skolemisation. In the presentation of the background theory in Fig. 1 quantifiers have already been eliminated.

The $\mathcal{FO}(\mathrm{K}(\mathsf{E})_n)$-definitions for the connectives are easily obtained from (1) by replacing $\mathfrak{M}, x \models \varphi$ by $\nu(\varphi, x)$ and meta-linguistic connectives by $\mathcal{FO}(\mathrm{K}(\mathsf{E})_n)$-connectives, which are shown in Fig. 2. All formulae are universally quantified. In order to obtain tableau rules for the connectives, we divide S^0 into two disjoint sets of formulas (S^0_+ and S^0_-) and write them in implicational form as given in Fig. 3. Obtaining S^0_- involves taking the contra-

$$
\begin{array}{ll}
S^0_+ : & \nu(\neg\varphi, x) \rightarrow \neg\nu(\varphi, x) \\
& \nu(\varphi \wedge \psi, x) \rightarrow \nu(\varphi, x) \wedge \nu(\varphi, x) \\
& \nu(\Diamond\varphi, x) \rightarrow (R(x, f(\varphi, x)) \wedge \nu(\varphi, f(\varphi, x))) \\
& \nu(\mathsf{E}_{>n}\varphi, x) \rightarrow (\bigwedge_{0<i\leq n+1} \nu(\varphi, f_i(\varphi, x)) \wedge \bigwedge_{0<i<j\leq n+1} (f_i(\varphi, x) \not\approx f_j(\varphi, x))) \\
S^0_- : & \neg\nu(\neg\varphi, x) \rightarrow \nu(\varphi, x) \\
& \neg\nu(\varphi \wedge \psi, x) \rightarrow (\neg\nu(\varphi, x) \vee \neg\nu(\psi, x)) \\
& \neg\nu(\Diamond\varphi, x) \rightarrow (\neg R(x, y) \vee \neg\nu(\varphi, y)) \\
& \neg\nu(\mathsf{E}_{>n}\varphi, x) \rightarrow (\bigvee_{0<i\leq n+1} \neg\nu(\varphi, y_i) \vee \bigvee_{0<i<j\leq n+1} (y_i \approx y_j))
\end{array}
$$

Figure 3: Sets S^0_+ and S^0_- of Skolemised rules for connectives

Rules for the connectives:

$$
(\neg)\ \frac{\nu(\neg\varphi, x)}{\neg\nu(\varphi, x)} \quad (\neg\neg)\ \frac{\neg\nu(\neg\varphi, x)}{\nu(\varphi, x)} \quad (\wedge)\ \frac{\nu(\varphi \wedge \psi, x)}{\nu(\varphi, x), \nu(\psi, x)} \quad (\neg\wedge)\ \frac{\neg\nu(\varphi \wedge \psi, x)}{\neg\nu(\varphi, x) \mid \neg\nu(\psi, x)}
$$

$$
(\Diamond)\ \frac{\nu(\Diamond\varphi, x)}{R(x, f(\Diamond\varphi, x)), \nu(\varphi, f(\Diamond\varphi, x))} \qquad (\neg\Diamond)\ \frac{\neg\nu(\Diamond\varphi, x), y \approx y}{\neg R(x, y) \mid \neg\nu(\varphi, y)}
$$

$$
(\mathsf{E}_{>n})\ \frac{\nu(\mathsf{E}_{>n}\varphi, x)}{\nu(\varphi, f_1(\mathsf{E}_{>n}\varphi, x), \cdots, \nu(\varphi, f_{n+1}(\mathsf{E}_{>n}\varphi, x)) \underset{0<i<j\leq n+1}{,} f_i(\mathsf{E}_{>n}\varphi, x) \not\approx f_j(\mathsf{E}_{>n}\varphi, x)}
$$

$$
(\neg\mathsf{E}_{>n})\ \frac{\neg\nu(\mathsf{E}_{>n}\varphi, x), y_1 \approx y_1, \ldots, y_{n+1} \approx y_{n+1}}{\neg\nu(\varphi, y_1) \mid \cdots \mid \neg\nu(\varphi, y_{n+1}) \underset{0<i<j\leq n+1}{\mid} (y_i \approx y_j)}
$$

Rules for equality:

$$
\frac{\nu(\varphi, x)}{x \approx x} \qquad \frac{x \approx y}{y \approx x} \qquad \frac{x \approx x, y \approx y, z \approx z}{x \not\approx y \mid y \not\approx z \mid x \approx z}
$$

$$
\frac{x \approx y, R(x, z)}{R(y, z)} \qquad \frac{x \approx y, R(z, x)}{R(z, y)} \qquad \frac{x \approx y, \nu(\varphi, x)}{\nu(\varphi, y)} \qquad \frac{x \approx y}{f(\varphi, x) \approx f(\varphi, y)}
$$

Closure rules:

$$
\frac{\nu(\varphi, x), \neg\nu(\varphi, x)}{\bot} \qquad \frac{R(x, y), \neg R(x, y)}{\bot} \qquad \frac{x \approx y, x \not\approx y}{\bot}
$$

Figure 4: The rules of $\mathcal{T}_{\mathsf{K}(\mathsf{E})_n}$

positives of the right-to-left implications of the formulae in S^0, turning them into prenex normal form and eliminating quantifiers via Skolemisation. The specification in Fig. 3 is *normalised* as defined in [15]. Note that the rule for $\mathsf{E}_{>n}$ in S^0 is in fact a scheme with n as a parameter. It follows that it generates an infinite set Ξ of actual $\mathsf{E}_{>n}$ formulae, in principle causing an infinite blow-up of S^0. However, Ξ is recursively enumerable so it can be turned into one tableau rule for $\mathsf{E}_{>n}$.

3 The tableau calculus $\mathcal{T}_{\mathbf{K}(\mathsf{E})_n}$

The tableau calculus synthesised from the semantic specification provided in the previous section in Figs. 1 and 3 is given in Fig. 4. We refer to it as $\mathcal{T}_{\mathsf{K}(\mathsf{E})_n}$.

Intuitively, an expression of the form $\nu(\varphi, x)$ means that a formula φ holds in a world x, and $\neg\nu(\varphi, x)$ means the contrary. If a rule is a multi-conclusion rule, we enumerate in the denominator all conclusions using commas. If rule is branching, we write in the denominator all alternative conclusions, separating them with | signs. Given Γ, the input set of formulas, we start the derivation putting in the initial node of the tableau all formulas from the set $\{\nu(\varphi, x) \mid \varphi \in \Gamma\}$ where x is a fresh constant of the domain sort.

The tableau rules for the Boolean connectives are quite straightforward. However, two different negation signs occur in the rule $(\neg)$, the one in the premise denoting $\mathrm{K}(\mathsf{E})_n$-negation, and the one occurring in the conclusion denoting $\mathcal{FO}(\mathrm{K}(\mathsf{E})_n)$-negation.

In the positive rules for $\Diamond$ and $\mathsf{E}_{>n}$ after eliminating existential quantifiers, instead of introducing a fresh constant, a Skolem term of the form $f(\varphi, x)$ is used. The advantage of this approach is fewer rule applications during the derivation. If, for example, at a certain point in a derivation we obtain the formula $x \approx f(\varphi, x)$, by the appropriate equality rules we also automatically obtain $x \approx f(\varphi, f(\varphi, x))$, whereas this would not be possible if we use fresh constants.

In this version of the tableau calculus the $(\neg\Diamond)$ rule is a branching rule but in Section 6 we prove that it is possible to turn the $\neg R(x, y)$ conclusion into a premise and thus decrease branching. The occurrence of the $y \approx y$ formula in the $(\neg\Diamond)$ rule explicitly ensures that the rule is applied only to the worlds that occur in the current branch (and to them only).

The $(\mathsf{E}_{>n})$ rule reflects the semantics of the global counting operator. It is generic, since applied to $\mathsf{E}_{>n}\varphi$, it produces $n + 1$ new, distinct worlds in which φ holds. The intuitive meaning of $(\neg\mathsf{E}_{>n})$ is as follows. Since it is not the case that there are more than n worlds in which φ holds, there are at most n worlds in which φ holds. So, for each $n+1$-tuple of worlds that have already appeared in the branch either some of them do not satisfy φ or some of them are equal. Again, we use $y \approx y$ formulas to ensure the rule is applied only to domain terms occurring in the current branch. It follows that $(\neg\mathsf{E}_{>n})$ is not applicable until at least $n+1$ worlds occur in the branch. We show in Section 6, the rule $(\neg\mathsf{E}_{>n})$, contrary to the rule $(\neg\Diamond)$, is not refinable by turning some of its conclusions to premises.

The first three rules for equality express the fact that it is an equivalence relation. Note that the synthesis of the reflexivity rule departs from the normal scheme for formulas in the background theory. The reason is that formulas $x \approx x$ express occurrences of a world x in a branch. If we left the premise of this rule empty, it would allow for the unconstrained creation of new worlds in a branch, which could affect applications of the $(\neg\Diamond)$ and $(\neg\mathsf{E}_{>n})$ rules. The remaining four rules state congruence of $\approx$ with respect to ν and R predicates and Skolem functions.

The closure rules are self-evident and require no explanation.

4 Soundness and completeness of $\mathcal{T}_{\mathbf{K}(\mathsf{E})_n}$

In this section we derive soundness and completeness of $\mathcal{T}_{\mathrm{K}(\mathsf{E})_n}$, by using results established in [15]. The appendix gives an independent proof of constructive completeness, and thus completeness, of $\mathcal{T}_{\mathrm{K}(\mathsf{E})_n}$.

In general, a tableau calculus $\mathcal{T}$ is *sound* iff for each satisfiable input set of formulas Γ each tableau $\mathcal{T}(\Gamma)$ is open, i.e., there exists a fully expanded branch in which no closure rule was applied. A tableau calculus is called *complete* iff for each unsatisfiable input set of formulas Γ there exists a closed tableau, i.e. a tableau where a closure rule was applied in each branch.

Since each rule of the calculus preserves satisfiability, which can be proven by easy verification, we obtain:

Theorem 4.1 *The tableau calculus $\mathcal{T}_{\mathrm{K}(\mathsf{E})_n}$ is sound.*

In [15] three conditions for the *well-definedness* of the semantic specification were formulated and proven to be sufficient for completeness of a synthesised calculus. We show that these conditions are satisfied by the semantic specification from Figs. 1 and 3 and, therefore, that $\mathcal{T}_{\mathrm{K}(\mathsf{E})_n}$ is complete.

A semantic specification S is *well-defined* if it is *normalised* and satisfies the following conditions [15]:

(wd1) all first-order models characterized by S are also models for $S^0 \cup S^b$;

(wd2) the relation $\prec$ induced by S is a well-founded ordering on expressions of the object language;

(wd3) given a formula φ, its decomposition by a suitable S-rule should logically imply the decomposition of φ by appropriately instantiated suitable $S^0 \cup S^b$-rule in the class of models for $S^0 \cup S^b$.

Conditions (wd1) and (wd3) are straightforwardly satisfied because $S = S^0 \cup S^b$. The ordering imposed by S is a direct subexpression ordering on $\mathrm{K}(\mathsf{E})_n$-expressions:

$$\begin{gathered}\varphi \prec \neg\varphi,\ \varphi \prec \varphi \wedge \psi,\ \neg\varphi \prec \neg(\varphi \wedge \psi),\ \psi \prec \varphi \wedge \psi,\ \neg\psi \prec \neg(\varphi \wedge \psi),\\ \varphi \prec \Diamond\varphi,\ \neg\varphi \prec \neg\Diamond\varphi,\ \varphi \prec \mathsf{E}_{>n}\varphi,\ \neg\varphi \prec \neg\mathsf{E}_{>n}\varphi\end{gathered} \tag{2}$$

so is well-founded and thus satisfies (wd2). Then it follows from [15] that:

Theorem 4.2 *$\mathcal{T}_{\mathrm{K}(\mathsf{E})_n}$ is complete.*

5 Termination of the calculus

Due to the presence of global operators the calculus $\mathcal{T}_{\mathrm{K}(\mathsf{E})_n}$ can generate infinite derivations for a satisfiable input set of formulas. A good example of such input is the formula $\mathsf{E}_{=0}(\neg\Diamond p)$. No such problem occurs if the input set is unsatisfiable. The logic $\mathrm{K}(\mathsf{E})_n$ is embeddable into two-variable fragment of first order logic with counting quantifiers and is therefore compact. It means that if an input is unsatisfiable any $\mathcal{T}_{\mathrm{K}(\mathsf{E})_n}$-derivation eventually closes all branches. However, the presence of $\mathsf{E}_{>n}$-operators, which are universal modalities with numerical restrictions, can interfere with termination if a particular input set of formulas is satisfiable.

Let $\leq_{\mathcal{B}}$ be an order of occurrences of expressions in a branch $\mathcal{B}$ and let $[x]_{\approx} = \{y \mid x \approx y \in \mathcal{B}\}$. The *unrestricted blocking* mechanism consists of the following rule and two strategy conditions added to the calculus:

$$(\mathsf{ub})\ \ \frac{x \approx x, y \approx y}{x \approx y \mid x \not\approx y};$$

(c1) The (ub)-rule should be applied exhaustively in a branch $\mathcal{B}$ before each application of the $(\mathsf{E}_{>n})$ and $(\Diamond)$-rules;

(c2) If $\mathsf{E}_{>n}$- or $\Diamond$-formula φ occurs in $x_1, \ldots, x_n \in [x]_\approx$, then the $(\mathsf{E}_{>n})$ rule and $(\Diamond)$ rule, respectively, are only applied to the least element in $[x]_\approx$ with respect to $\leq_\mathcal{B}$.

Here, we assume that an input set of formulas Γ is finite. As a consequence of conditions (c1) and (c2) we obtain an easy observation that for each x occurring in $\mathcal{B}$ the $(\mathsf{E}_{>n})$ and $(\Diamond)$-rules can be applied to elements of $[x]_\approx$ only finitely many times, subject to the number of distinct $\mathsf{E}_{>n}$ and $\Diamond$-formulas in Γ. It follows that a fully expanded, open $\mathcal{T}^{(\mathsf{ub})}_{\mathrm{K}(\mathsf{E})_n}$-tableau branch $\mathcal{B}$ is finite iff $\mathfrak{M}(\mathcal{B})$ is finite. $\mathfrak{M}(\mathcal{B})$ is the model induced by any fully expanded, open branch $\mathcal{B}$. A formal definition is given in the Appendix.

Now we formulate a stronger result (cf. [16, Lemma 13]) that links arbitrary models and $\mathcal{T}^{(\mathsf{ub})}_{\mathrm{K}(\mathsf{E})_n}$-tableau for a satisfiable set of input formulas Γ:

Proposition 5.1 *Let Γ be any set of $\mathrm{K}(\mathsf{E})_n$-formulas. Suppose that $\mathfrak{N} = \langle \mathbf{U}, \mathbf{S}, \mathbf{Z} \rangle$ is a model for Γ. Then, there exists a branch $\mathcal{B}$ in $\mathcal{T}^{(\mathsf{ub})}_{\mathrm{K}(\mathsf{E})_n}$-tableau for Γ such that $\mathsf{Card}(\mathbf{U}) \geq \mathsf{Card}(\mathcal{B})$, where $\mathsf{Card}(\mathcal{B}) = \mathsf{Card}(\{[x]_\approx \mid x \approx x \in \mathcal{B}\})$.*

Proposition 5.1 states that if an input set Γ has a finite model then any $\mathcal{T}^{(\mathsf{ub})}_{\mathrm{K}(\mathsf{E})_n}$-tableau derivation will be terminating. We say that a tableau calculus $\mathcal{T}$ is *terminating* (for satisfiability) iff for every finite set of formulas Γ every closed tableau $\mathcal{T}(\Gamma)$ is finite and every open tableau $\mathcal{T}(\Gamma)$ has a finite open branch. To ensure termination for $\mathcal{T}^{(\mathsf{ub})}_{\mathrm{K}(\mathsf{E})_n}$ it suffices therefore to establish the finite model property for $\mathrm{K}(\mathsf{E})_n$, which we do next.

Theorem 5.2 (Finite Model Property) *The logic $\mathrm{K}(\mathsf{E})_n$ has the effective finite model property with the bounding function $\mu = 2^{\mathsf{Card}(\{\mathrm{Sub}(\varphi)\})+\log(n+1)}$ for any given input formula φ, where $\mathrm{Sub}(\varphi)$ is the set of all subformulas of φ and $n = \max\{m : \mathsf{E}_{>m}\psi \in \mathrm{Sub}(\varphi)\}$, where n is coded in binary, i.e., whenever φ has a model, it also has a model of the size not exceeding $2^{\mathsf{Card}(\{\mathrm{Sub}(\varphi)\})+\log(n+1)}$.*

As a consequence of Proposition 5.1, Theorem 5.2 and the fact that each rule of $\mathcal{T}^{(\mathsf{ub})}_{\mathrm{K}(\mathsf{E})_n}$ is finitely branching we get:

Theorem 5.3 *$\mathcal{T}^{(\mathsf{ub})}_{\mathrm{K}(\mathsf{E})_n}$ is terminating.*

We also have:

Theorem 5.4 *The logic $\mathrm{K}(\mathsf{E})_n$ is* NEXPTIME*-complete.*

Membership of $\mathrm{K}(\mathsf{E})_n$ in NEXPTIME follows from Theorem 5.2 (see also [13]). Hardness follows from the fact that it is possible to encode a finite tiling problem within $\mathrm{K}(\mathsf{E})_n$ (see [9]).

To provide a complexity-optimal derivation strategy for $\mathcal{T}^{(\mathsf{ub})}_{\mathrm{K}(\mathsf{E})_n}$ we formulate the following condition:

(op) Expand a branch of a $\mathcal{T}^{(\mathsf{ub})}_{\mathrm{K}(\mathsf{E})_n}$-tableau until the number of equivalence classes of worlds in $\mathcal{B}$ exceeds the bound from Theorem 5.2. Then stop.

Complexity-optimality of any $\mathcal{T}^{(\mathsf{ub})}_{\mathrm{K}(\mathsf{E})_n}$-derivation using this strategy is a simple consequence of Proposition 5.1 and Theorems 5.2 and 5.4.

6 Rule refinement of $\mathcal{T}_{\mathbf{K}(\mathsf{E})_n}$

In [15] cases are presented where the rules of synthesised tableau calculi can be refined. Sometimes we can refine a calculus by decreasing the *branching factor* of particular rules, *ipso facto* reducing size of a tableau. In order to do that, we turn some conclusions of a branching rule into its premises and inverting negation in front of them. More precisely, let β be a tableau rule $X_0/X_1 \mid \cdots \mid X_n$ and $X_j = \{\chi_1, \ldots, \chi_m\}$. *Refinements* of the rule β with respect to its j-th branch are the rules β^j_k for $k = 1, \ldots, m$ defined by

$$\beta^j_k = \frac{X_0, \sim\chi_k}{X_1 \mid \cdots \mid X_{j-1} \mid X_{j+1} \mid \cdots \mid X_n}.$$

Here, $\sim\varphi = \psi$ if $\varphi = \neg\psi$, and $\sim\varphi = \neg\varphi$ otherwise. A *refinement* $\mathcal{T}^R$ of a given calculus $\mathcal{T}$ is obtained by replacing a rule β with its refinements $\beta^j_1, \ldots, \beta^j_m$.

Notwithstanding the conceptual simplicity of the foregoing method, not every branching rule can be refined without loosing completeness for the whole calculus. Furthermore, it might occur that even the same branching rule behaves differently in different tableau calculi allowing for refinement in one case but not in another.

It is straightforward that refinements of a rule are derivable in any given calculus. Hence, every derivation step in the refined calculus can be simulated in the original calculus. To obtain the converse and, thus, equivalence of the refined and original calculi we scrutinise the notion of *admissibility* of the original rule in the refined calculus. Hence, we need to provide the condition that would help to decide whether a rule β is admissible in a calculus $\mathcal{T}^R$. In [15] we can find such a condition:

Theorem 6.1 ([15]) *Let $\mathcal{B}$ is an arbitrary open and fully expanded branch in a $\mathcal{T}^R$-tableau. Let $F = \{\varphi_1, \ldots, \varphi_l\}$ be a set of all $\mathrm{K}(\mathsf{E})_n$-formulas from $\mathcal{B}$ reflected in $\mathfrak{M}(\mathcal{B})$. Then the branch $\mathcal{B}$ is reflected in $\mathfrak{M}(\mathcal{B})$ provided the following condition (†) is satisfied: If $X_0(\varphi_{i_1}, \ldots, \varphi_{i_k}) \in \mathcal{B}$ then $\mathfrak{M}(\mathcal{B}) \models X_m(\varphi_{i_1}, \ldots, \varphi_{i_k})$, for some $m \in \{1, \ldots, n\}$.*

The notions of *reflecting a branch by a model* and *constructive completeness* are explained in the Appendix. In particular, the (†) condition says that any hypothetical application of the rule β is redundant in the branch $\mathcal{B}$ of the refined calculus. As a corollary, we obtain the following statement.

Theorem 6.2 ([15]) *If $\mathcal{T}$ is constructively complete and the (†) condition*

Refined $(\neg\Diamond)$-rule:

$$(\neg\Diamond)\ \frac{\neg\nu(\Diamond\varphi, w), z \approx z}{\neg R(w,z) \mid \neg\nu(\varphi, z)} \quad \rightsquigarrow \quad (\neg\Diamond)^R\ \frac{\neg\nu(\Diamond p, w), R(w,z)}{\neg\nu(p,z)}$$

Refined transitivity rule for equality:

$$(\mathrm{Tr})\ \frac{}{x \not\approx y \mid y \not\approx z \mid x \approx z} \quad \rightsquigarrow \quad (\mathrm{Tr})^R\ \frac{x \approx y, y \approx z}{x \approx z}$$

Figure 5: Refined rules of $\mathcal{T}^R_{\mathbf{K}(\mathsf{E}_n)}$

holds for every open and fully expanded branch in a $\mathcal{T}^R$-tableau, then $\mathcal{T}^R$ is constructively complete.

In $\mathcal{T}_{\mathbf{K}(\mathsf{E}_n)}$ two branching rules are refinable by decreasing their branching factor: the $(\neg\Diamond)$ rule and the transitivity rule for equality (as shown in Fig. 5).

Proposition 6.3 *Condition (†) is satisfied in $\mathcal{T}_{\mathbf{K}(\mathsf{E}_n)}$ for the rules $(\neg\Diamond)^R$ and $(Tr)^R$ from Figure 5.*

Refining any of the remaining branching rules turns $\mathcal{T}_{\mathbf{K}(\mathsf{E}_n)}$ into an incomplete calculus. As it was mentioned above, the rules $(\neg\wedge)$ and $(\neg\mathsf{E}_{>n})$ are not refinable. An example showing that $(\neg\wedge)$ cannot be refined can be obtained from a similar example for the disjunction rule in [15]. Due to the large branching factor of the $(\neg\mathsf{E}_{>n})$ rule, its unrefinability to $\frac{\neg\nu(\mathsf{E}_{>n}\varphi,x),\neg\nu(\varphi,y_1),\ldots,\neg\nu(\varphi,y_{n+1})}{\bigvee_{0<i<j\leq n+1}(y_i\approx y_j)}$ is a more significant loss. As an example consider the input set $\Gamma = \{\nu(\neg\mathsf{E}_{>1}\neg p, x), \nu(q,y)\}$. No rules are applicable to Γ therefore the derivation stops at the initial node. A model reflecting the branch $\mathfrak{M}(\mathcal{B}) = \langle \mathbf{W}, \mathbf{R}, \mathbf{v}\rangle$ is defined by: $\mathbf{W} = \{\{x\},\{y\}\}$, $\mathbf{R} = \varnothing$, $\mathbf{v}(p) = \varnothing$, $\mathbf{v}(q) = \{y\}$. It therefore follows that both worlds x and y satisfy $\neg p$, so $\neg\mathsf{E}_{>1}\neg p$ is not satisfied by $\mathfrak{M}(\mathcal{B})$ although the formula is evidently satisfiable.

Both failures could be neutralized by introducing an analytic cut rule: $\frac{x\approx x}{\nu(\varphi,x)\mid\neg\nu(\varphi,x)}$ where x is any term of the domain sort and φ is a subformula of a formula occurring in the branch. The rule is the tableau-counterpart to the law of excluded middle.

7 Refinement via elimination of domain sort symbols

A second refinement described in [15] allows the synthesised rules to the reformulated without the use of the ν-predicate or any other domain sort symbols. This is possible for logics allowing the encoding of domain expressions by its own means. This can be done for hybrid logic with the @-operator. There are three basic cases of domain expressions (modulo negation): $\nu(\varphi,x)$, $R(x,y)$, $x \approx y$. Each of them can be mimicked by a suitable hybrid logic $\mathcal{H}(@)$-formula [15]:

$$\begin{array}{ll} \mathcal{H}(\nu(\varphi,x)) = @_{i_x}\varphi & \mathcal{H}(\neg\nu(\varphi,x)) = @_{i_x}\neg\varphi \\ \mathcal{H}(R(x,y)) = @_{i_x}\Diamond j_y & \mathcal{H}(\neg R(x,y)) = @_{i_x}\neg\Diamond j_y \\ \mathcal{H}(x\approx y) = @_{i_x} j_y & \mathcal{H}(x\not\approx y) = @_{i_x}\neg j_y \end{array} \tag{3}$$

$\mathbf{K}(\mathsf{E}_n)(\nu(\varphi, x)) = \mathsf{E}_{=1}p_x \wedge \mathsf{E}_{=0}(p_x \wedge \neg\varphi)$
$\mathbf{K}(\mathsf{E}_n)(\neg\nu(\varphi, x)) = \mathsf{E}_{=1}p_x \wedge \mathsf{E}_{=0}(p_x \wedge \varphi)$
$\mathbf{K}(\mathsf{E}_n)(R(x, y)) = \mathsf{E}_{=1}p_x \wedge \mathsf{E}_{=1}p_y \wedge \mathsf{E}_{=0}(p_x \wedge \neg\Diamond p_y)$
$\mathbf{K}(\mathsf{E}_n)(\neg R(x, y)) = \mathsf{E}_{=1}p_x \wedge \mathsf{E}_{=1}p_y \wedge \mathsf{E}_{=0}(p_x \wedge \Diamond p_y)$
$\mathbf{K}(\mathsf{E}_n)(x \approx y) = \mathsf{E}_{=1}p_x \wedge \mathsf{E}_{=1}p_y \wedge \mathsf{E}_{=0}(p_x \wedge \neg p_y)$
$\mathbf{K}(\mathsf{E}_n)(x \not\approx y) = \mathsf{E}_{=1}p_x \wedge \mathsf{E}_{=1}p_y \wedge \mathsf{E}_{=0}(p_x \wedge p_y)$
In each case p_x, p_y are fresh propositional variables not occurring in Γ.

Figure 6: $\mathrm{K}(\mathsf{E})_n$-expressions to replace domain sort expressions

The logic $\mathrm{K}(\mathsf{E})_n$ is not equipped with syntactic constructs for naming worlds such as nominals in hybrid logic $\mathcal{H}(@)$. However, due to the fact that the $\mathsf{E}_{>n}$-operators combine counting properties and global range, we are able to bypass explicit labelling expressions by exploiting the $\mathsf{E}_{=1}$-operator in an appropriate way. For each input set of formulas Γ we can encode domain sort expressions as specified in Fig. 6. We only need to show that the respective formulas in the figure actually encode the respective domain sort expressions.

Proposition 7.1 *The domain sort expressions $\nu(\varphi, x)$, $R(x, y)$, $x \approx y$ hold in a model $\mathcal{I}$ iff the respective $\mathbf{K}(\mathsf{E}_n)$-formulas from Fig. 6 hold in a suitable conservative extension $\mathfrak{M}'$ of a Kripke model $\mathfrak{M} = \langle W, R, V\rangle$, where $W = \mathit{Dom}(\mathcal{I})$, $R^{\mathcal{I}} = R^{\mathfrak{M}}$ and $\nu(p, x)$ iff $x \in V(p)$ for each $p \in$ PROP.*

For convenience and economy of space, we introduce a new *colon* notation. We abbreviate formulas of the form: $\mathsf{E}_{=1}\varphi \wedge \mathsf{E}_{=0}(\varphi \wedge \neg\psi)$ as $\varphi : \psi$.

To accomplish the refinement of our calculus we need to introduce one additional element. We define a countable set $\mathcal{F} = \{f_j\}_{j\in\mathbb{N}}$ where each f_j : FORM $\times$ FORM $\rightarrow$ FORM is a function. The intention is to have a countable set of function symbols to obtain an expression class analogous to a class of Skolem terms in the first-order meta-language.

Figure 7 presents the rules of the refined calculus $\mathcal{T}^{R_d}_{\mathrm{K}(\mathsf{E})_n}$ including the unrestricted blocking mechanism. A benefit resulting from dispensing with domain symbols is fewer rules.

The *colon* notation makes the refined tableau calculus resemble standard prefixed calculi. For a given input set of formulas Γ a label of the initial node is a propositional variable obtained by translating $\nu(\bigwedge\Gamma, x)$ based on the encoding in Fig. 6. New labels introduced by the rules ($\Diamond$) and ($\mathsf{E}_{>n}$) are arbitrary formulas (by definition of functional symbols f_i). The reader might be surprised by the rule ($\Diamond$), since the first formula in the conclusion is of the same form as the premise formula, which might in principle lead to infinite derivations. What distinguishes these two is the fact that the formula under the scope of $\Diamond$ in the premise is not necessarily a labelling formula, whereas a formula that appears under the scope of $\Diamond$ in the conclusion certainly is. This makes it subject to applications of the ($\mathsf{sub1}$), ($\mathsf{sub2}$) and (ub) rules which ensure finiteness (with respect to ($\Diamond$)-application) of at least one of the branches.

The refinement consisting of refining away domain sort symbols, as de-

Rules for the connectives:

$$(\neg\neg)\ \frac{\varphi : \neg\neg\psi}{\varphi : \psi} \qquad (\wedge)\ \frac{\varphi : \psi \wedge \chi}{\varphi : \psi, \varphi : \chi} \qquad (\neg\wedge)\ \frac{\varphi : \neg(\psi \wedge \chi)}{\varphi : \neg\psi \mid \varphi : \neg\chi}$$

$$(\Diamond)\ \frac{\varphi : \Diamond\psi}{\varphi : \Diamond f(\Diamond\psi, \varphi), f(\Diamond\psi, \varphi) : \psi} \qquad (\neg\Diamond)\ \frac{\varphi : \neg\Diamond\psi, \varphi : \Diamond\chi, \chi : \chi}{\varphi : \neg\psi}$$

$$(\mathsf{E}_{>n})\ \frac{\varphi : \mathsf{E}_{>n}\psi}{f_1(\mathsf{E}_{>n}\psi, \varphi) : \psi, \ldots, f_{n+1}(\mathsf{E}_{>n}\psi, \varphi) : \psi \underset{0<k<l\leq n+1}{,} f_k(\mathsf{E}_{>n}\psi, \varphi) : \neg f_l(\mathsf{E}_{>n}\psi, \varphi)}$$

$$(\neg\mathsf{E}_{>n})\ \frac{\varphi : \neg\mathsf{E}_{>n}\psi, \chi_1 : \chi_1, \ldots, \chi_{n+1} : \chi_{n+1}}{\chi_1 : \neg\psi \mid \cdots \mid \chi_{n+1} : \neg\psi \underset{0<k<l\leq n+1}{\mid} \chi_k : \neg\chi_l}$$

Rules for equality, closure rule, unrestricted blocking rule:

$$(\mathsf{ref})\ \frac{\varphi : \psi}{\varphi : \varphi} \qquad (\mathsf{con})\ \frac{\varphi : \psi, \psi : \psi}{f(\chi, \varphi) : f(\chi, \psi)} \qquad (\mathsf{sub1})\ \frac{\varphi : \psi, \psi : \psi, \varphi : \chi}{\psi : \chi}$$

$$(\bot)\ \frac{\varphi : \psi, \varphi : \neg\psi}{\bot} \qquad (\mathsf{ub})\ \frac{\varphi : \varphi, \psi : \psi}{\varphi : \psi \mid \varphi : \neg\psi} \qquad (\mathsf{sub2})\ \frac{\varphi : \psi, \psi : \psi, \chi : \theta}{\chi : \theta[\psi/\varphi]}$$

Figure 7: Rules for refined calculus $\mathcal{T}^{R_d}_{\mathrm{K}(\mathsf{E})_n}$

scribed, preserves soundness, constructive completeness and termination. In order to conduct proof of completeness, we define the 'branch induced model' $\mathfrak{M}(\mathcal{B}) = \langle \mathbf{W}, \mathbf{R}, \mathbf{V}\rangle$ as follows. Let $\varphi \sim_{\mathcal{B}} \psi$ iff $\varphi : \psi, \psi : \psi \in \mathcal{B}$. Then:

$\mathbf{W} = \{[\varphi]_{\sim_{\mathcal{B}}} \mid \varphi : \varphi \in \mathcal{B}\}$; $\mathbf{R} = \{([\varphi]_{\sim_{\mathcal{B}}}, [\psi]_{\sim_{\mathcal{B}}}) \mid \varphi : \Diamond\psi, \psi : \psi \in \mathcal{B}\}$;
$\mathbf{V} = \{(p, U) \mid p \in \textsc{prop}, p \text{ occurs in } \mathcal{B} \text{ and } U = \{[\varphi]_{\sim_{\mathcal{B}}} \mid \varphi : p \in \mathcal{B}\}\}$.

It follows from the construction of $\mathcal{T}^{R_d}_{\mathrm{K}(\mathsf{E})_n}$-tableau that we do not need to interfere with the well-founded ordering given in (2).

8 MetTeL2 implementation

In order to implement a prover based on the prefix tableau calculus $\mathcal{T}_{\mathrm{K}(\mathsf{E})_n}$ we used MetTeL2 [17,1], a new tableau prover generator. MetTeL2 automatically generates JAVA code of a tableau prover from the syntax specification of a logical theory and a set of tableau rules provided by the user. MetTeL2 fully supports dynamic backtracking and backjumping and uses the unrestricted blocking mechanism to ensure termination.

At the moment, MetTeL2 does not support specifications of logical connectives and tableau rules which are parameterised by numerical values. Because of this, all the operators $\mathsf{E}_{>n}$ for different n are specified as separate connectives up to some fixed number N. For each particular number N required for the formalisation of a set of test problems, we wrote a script to automatically produce a language specification and a tableau specification for the logic $\mathrm{K}(\mathsf{E})_n$, where the parameter n in the operators $\mathsf{E}_{>n}$ is limited by N. The obtained language and tableau specifications are passed to MetTeL2 which generates a tableau prover for the logic of the restricted language.

The generated prover is not intended to be a state-of-the-art prover but can be used for experimenting with the tableau calculus. During our investigation we found it useful to test problems and experiment with different rule refinements. For example, experimenting with the $(\neg\mathsf{E}_{>n})$ rule and its refined variant we discovered that the calculus loses completeness under such a refinement. This directed us to look for a counterexample to the condition (†) and to realise subsequently that this rule is not refinable. Various further experiments showed that it is possible to eliminate nominals from the calculus replacing them by formulae without loosing good properties of the calculus. The generated provers can be optimised and tuned further by integrating user implemented proof strategies using the of provided API of MetTeL2.

A specifications of the syntax of $\mathrm{K}(\mathsf{E})_n$ and the presented tableau calculus are available at the MetTeL2 on-line demo page [1], where the user is allowed to amend the specifications and to regenerate the prover.

9 Related work

Several tableau-based decision procedures for logics with counting operators have been established so far, all of them in the field of description logics. We mention three to draw some parallels between them and our system.

Kaminski, Schneider and Smolka in [11] introduce a tableau algorithm for the logic $\mathcal{SHOQ}$ (a description logic with nominals, graded modalities and graded counting modalities). The tableau rules for the counting fragment resemble ours, the only difference lies in a blocking mechanism they use, namely pattern-based blocking. It consists in searching potential successors of worlds among worlds that are already present in a branch. In this search the authors exploit the notion of pattern which is a set of $\Diamond$-formulas and $\Box$-formulas that hold in a particular world. Advantages of this approach are: keeping the tableau calculus in the complexity class of the logic, namely it is NexpTime-complete, and ensuring fewer mergings of worlds. Compared to our calculus the approach is conceptually more complex (apart from the notion of pattern itself, it introduces several new concepts such as evidence, quasi-evidence, evidence-completion). That makes the decision strategy quite complicated, though in terms of complexity the system of Kaminski et al. is complexity-optimal. This tableau calculus is internalised, i.e., it does not involve any extra-logical expressions to label the worlds in a tableau, however, it exploits nominals not present in $\mathrm{K}(\mathsf{E}_n)$. In the second refinement of our calculus we show how to dispose of meta-language domain expressions and internalise the semantics, only using global counting and not introducing any additional expressions.

Horrocks, Sattler and Tobies in [10] introduce a tableau algorithm for the logic $\mathcal{SHIQ}$ (a description logic with nominals, cardinality constraints and inverse operators). As for [11], the tableau rules for the counting operators are similar to ours, though we use Skolem terms whereas in [10] the rules introduce fresh constant. An aspect that distinguishes their tableau approach from ours is, again, the blocking technique involved. The mechanism they use is pairwise blocking. It consists in comparing not worlds but pairs of worlds. Assume that we have two pairs of worlds x, y and x', y'. The latter pair can be blocked iff the former pair occurred in a branch as first and $\mathcal{L}(x) = \mathcal{L}(x')$, $\mathcal{L}(y) = \mathcal{L}(y')$ and $\mathcal{L}(x, y) = \mathcal{L}(x', y')$. It follows that not only must the respective nodes satisfy the same formulas, but also they must be linked in the same way. Horrocks et al. thus obtain a terminating tableau algorithm. Their approach, again, is internalised but uses nominals as labels.

In [6] Faddoul, Farsina, Haarslev and Möller present a completely different tableau-based decision procedure for logics with cardinality constraints. They designed a hybrid system for the logic $\mathcal{ALCQ}$ (a description logic with cardinality constraints). Based on the atomic decomposition technique [12], they split the derivation process into an arithmetical part and a logical part. Boldly speaking, for each decided formula its 'counting' part (i.e., containing counting operators) is turned into a set of inequalities which has to be solved before the application of other rules to it. This approach was proven to be more efficient than the one presented in [10] but remains in the same complexity class (NexpTime-complete). Unfortunately, the atomic decomposition technique is not applicable to the logic $\mathrm{K}(\mathsf{E})_n$ since it requires role hierarchies not available in $\mathrm{K}(\mathsf{E})_n$.

In comparison to the aforementioned approaches, the approach presented in this paper is conceptually simpler due to the intuitive concept of unrestricted blocking incorporated into the calculus as an inference rule. Moreover, even though the (ub) rule allows for comparing arbitrary worlds present in a branch, the whole calculus remains complexity-optimal (which mainly follows from high computational complexity of $\mathrm{K}(\mathsf{E})_n$). We also showed how to encode the semantics of the logic, only using counting operators, and thus obtain a refined version of the calculus.

10 Concluding remarks

The presentation of tableau calculi for the logic $\mathrm{K}(\mathsf{E})_n$ in this paper shows that tableau calculi can be derived and refined in a systematic way based on the principles of the tableau synthesis framework [15]. We proved the finite model property of $\mathrm{K}(\mathsf{E})_n$, thus giving us an easy way to obtain tableau decision procedures by using the unrestricted blocking mechanism. Novel in this paper is the refinement that results in a 'direct' tableau calculus in which the rules are defined in the language of the object logic. This is possible as $\mathrm{K}(\mathsf{E})_n$ is expressive enough to define its own semantics.

Although in our considerations we confined ourselves to the logic K with global counting operators, it is apparent that no particular feature distinguishes logic K from other normal modal logics in respect to the results established in the paper. In fact, we can extend their scope by simply enriching the background theory with frame conditions for respective logics and slightly modifying proofs of completeness and termination theorems.

Possible directions of a future work involve specialising the unrestricted blocking mechanism so it deals with counting operators more efficiently. Furthermore, the efficiency of the hybrid approach of [6] is motivation to explore whether integer programming methods are applicable to $\mathrm{K}(\mathsf{E}_n)$ without introducing role hierarchies.

Acknowledgements

We thank Ian Pratt-Hartmann for valuable suggestions on Theorem 5.2. This research has been supported financially by the National Science Centre of Poland (decision no. DEC-2011/01/N/HS1/01979) and the UK EPSRC (grant no. EP/H043748/1).

References

[1] MetTeL *website*, `http://mettel-prover.org`.

[2] Areces, C., G. Hoffmann and A. Denis, *Modal logics with counting*, in: *Proceedings of WoLLIC 2010*, Brasilia, Brazil, 2010.

[3] Blackburn, P., M. de Rijke and Y. Venema, "Modal logic," Camb. Univ. Pr., NY, USA, 2001.

[4] Caro, F., *Graded modalities, II (canonical models)*, St. Log. **47** (1988), pp. 1–10.

[5] Cerrato, C., *Decidability by filtrations for graded normal logics (graded modalities v)*, St. Log. **53** (1994), pp. 61–73.

[6] Faddoul, J., N. Farsinia, V. Haarslev and R. Möller, *A hybrid tableau algorithm for alcq*, in: *Proceedings ECAI 2008* (2008), pp. 725–726.

[7] Fattorosi-Barnaba, M. and F. Caro, *Graded modalities. i*, St. Log. **44** (1985), pp. 197–221.

[8] Fine, K., *In so many possible worlds*, N. D. J. For. Log. **13** (1972), pp. 516–520.

[9] Fürer, M., *The computational complexity of the unconstrained limited domino problem*, in: *Proceedings of the Symposium "Rekursive Kombinatorik" on Logic and Machines: Decision Problems and Complexity* (1984), pp. 312–319.

[10] Horrocks, I., U. Sattler and S. Tobies, *Reasoning with individuals for the description logic shiq*, in: *Proceedings of (CADE-17)*, LNCS (2000).

[11] Kaminski, M., S. Schneider and G. Smolka, *Terminating tableaux for graded hybrid logic with global modalities and role hierarchies*, in: *TABLEAUX 2009*, LNCS (LNAI) **5607** (2009), pp. 235–249.

[12] Ohlbach, H. J. and J. Koehler, *Role hierarchies and number restrictions*, in: R. B. et. al., editor, *Description Logics*, URA-CNRS **410**, 1997.

[13] Pratt-Hartmann, I., *The two-variable fragment with counting revisited*, in: *Proceedings of WoLLIC'10*, 2010.

[14] Schmidt, R. A., *Synthesising terminating tableau calculi for relational logics: Invited paper*, in: *RAMiCS 12*, LNCS **6663** (2011), pp. 40–49.

[15] Schmidt, R. A. and D. Tishkovsky, *Automated synthesis of tableau calculi*, Log. Meth. in Comp. Sc. **7** (2011), pp. 1–32.

[16] Schmidt, R. A. and D. Tishkovsky, *Using tableau to decide description logics with full role negation and identity* (2011), manuscript, available at `http://www.mettel-prover.org/papers/ALBOid.pdf`.

[17] Tishkovsky, D., R. A. Schmidt and M. Khodadadi, MetTeL2*: Towards a prover generation platform (system description)* (2012), available at `http://www.mettel-prover.org/papers/MetTeL2SysDesc.pdf`.

[18] van der Hoek, W. and M. de Rijke, *Counting objects*, J. Log. Comp. **5** (1995), pp. 325–345.

[19] Zawidzki, M., *Adequacy of the logic* $\mathrm{K}(\mathsf{E}_n)$ (2011), to appear.

A Constructive completeness of $\mathcal{T}_{\mathbf{K}(\mathsf{E})_n}$

The calculus $\mathcal{T}_{\mathrm{K}(\mathsf{E})_n}$ is not only complete but also *constructively complete*. Before we give a formal definition of constructive completeness and prove that $\mathcal{T}_{\mathrm{K}(\mathsf{E})_n}$ has this property, we will introduce several preliminary notions.

For an open, fully expanded branch $\mathcal{B}$, let $\sim_{\mathcal{B}}$ be a relation defined as follows: $x \sim_{\mathcal{B}} y$ iff $x \approx y \in \mathcal{B}$. Because all the equality rules are applied in the branch, $\sim_{\mathcal{B}}$ is an equivalence relation.

For a model $\mathfrak{M} = (W, R, V)$ let $\gamma : \mathcal{FO}(\mathrm{K}(\mathsf{E})_n) \to W \cup \mathcal{FO}(\mathrm{K}(\mathsf{E})_n)$ be a function that maps each propositional variable of the sort 1 to itself and each term of the domain sort to an element of a domain W of a model $\mathfrak{M}$. We say that the model $\mathfrak{M}$ *reflects* a formula φ of the sort 1 that occurred in a branch $\mathcal{B}$ if the following two conditions are true.

(i) if $\nu(\varphi, x) \in \mathcal{B}$ then $\mathfrak{M}, \gamma(x) \models \varphi$;

(ii) if $\neg\nu(\varphi, x) \in \mathcal{B}$ then $\mathfrak{M}, \gamma(x) \not\models \varphi$.

We say that $\mathfrak{M}$ reflects $\mathcal{B}$ iff it reflects all formulas of the sort 1 that appear in $\mathcal{B}$.

It worth noting that, in general, $\nu(\varphi, x) \in \mathcal{B}$ and $\neg\nu(\varphi, x) \in \mathcal{B}$ are not complementary and there are tableau derivations where neither of such terms occur in some branch.

We call a tableau calculus $\mathcal{T}$ *constructively complete* iff for each input set of formulas Γ, for any open, fully expanded branch $\mathcal{B}$ in a tableau $\mathcal{T}(\Gamma)$ there exists a model $\mathfrak{M}$ such that:

(i) the domain W of $\mathfrak{M}$ is defined as follows: $W = \{[x]_{\sim_{\mathcal{B}}} \mid x \approx x \in \mathcal{B}\}$;

(ii) $\mathfrak{M}$ reflects $\mathcal{B}$ under the *canonical projection valuation* π defined as follows: $\pi(x) = [x]_{\sim_{\mathcal{B}}}$ for every term x of the domain sort that appeared in $\mathcal{B}$.

Now, suppose that Γ is a set of formulas. Let Γ be an input set for our tableau calculus $\mathcal{T}_{\mathrm{K}(\mathsf{E})_n}$. We denote a tableau for Γ by $\mathcal{T}_{\mathrm{K}(\mathsf{E})_n}(\Gamma)$. Let $\mathcal{B}$ be an open, fully-expanded branch of $\mathcal{T}_{\mathrm{K}(\mathsf{E})_n}(\Gamma)$. We associate with $\mathcal{B}$ a model $\mathfrak{M}(\mathcal{B}) = \langle \mathbf{W}, \mathbf{R}, \mathbf{V} \rangle$ defined as follows:

$$\begin{aligned}
\mathbf{W} &= \{\pi(x) \mid x \text{ is a term of the domain sort and } x \approx x \in \mathcal{B}\},\\
\mathbf{R} &= \{(\pi(x), \pi(y)) \mid x, y \text{ are terms of the domain sort and } R(x, y) \in \mathcal{B}\},
\end{aligned}$$

and $\mathbf{V}$ maps every propositional variable p of the sort 1 to the set $\{\pi(x) \mid \nu(p, x) \in \mathcal{B}\}$.

Lemma A.1 *Let Γ be an input set of $\mathrm{K}(\mathsf{E})_n$-formulas. Let $\mathcal{B}$ be an open, fully expanded branch of a tableau for Γ. Then $\mathfrak{M}(\mathcal{B})$ reflects $\mathcal{B}$, that is:*

(i) for any $\mathrm{K}(\mathsf{E})_n$-formula φ, if $\nu(\varphi, x) \in \mathcal{B}$ then $\mathfrak{M}(\mathcal{B}), \pi(x) \models \varphi$;

(ii) if $R(x, y) \in \mathcal{B}$ then $(\pi(x), \pi(y)) \in \mathbf{R}$;

(iii) if $x \approx y$ then $\pi(x) = \pi(y)$.

Proof Since (ii) and (iii) follow directly from the construction of $\mathbf{R}$, the definition of $\mathfrak{M}(\mathcal{B})$ and the construction of $\mathbf{W}$ respectively, we confine ourselves to the proof of (i). We proceed by induction on $\prec$ from (2).

$\varphi = p$. Since $\nu(p, x) \in \mathcal{B}$, by definition of $\mathbf{v}$ we obtain that $\pi(x) \in \mathbf{v}(p)$, whence $\mathfrak{M}(\mathcal{B}),\ \pi(x) \models p$.

$\varphi = \neg\psi$. We consider the following cases:

$\psi = \neg\chi$. We have that $\nu(\neg\neg\chi, x) \in \mathcal{B}$. Branch $\mathcal{B}$ is fully expanded so rules $(\neg)$ and $(\neg\neg)$ must have been already applied to $\varphi = \neg\neg\chi$. Hence $\nu(\chi, x) \in \mathcal{B}$. By induction hypothesis $\mathfrak{M}(\mathcal{B}), \pi(x) \models \chi$.

$\psi = \chi \wedge \theta$. $\nu(\neg(\chi\wedge\theta), x)$ occurred in $\mathcal{B}$. Because $\mathcal{B}$ is fully expanded we have that $(\neg\wedge)$ has been applied to $\nu(\neg(\chi \wedge \theta), x)$. Thus, either $\nu(\neg\chi, x) \in \mathcal{B}$ or $\nu(\neg\theta, x) \in \mathcal{B}$ holds. Suppose that the former is the case. Then, by induction hypothesis, $\mathfrak{M}(\mathcal{B}), \pi(x) \models \neg\chi$. It follows that $\mathfrak{M}(\mathcal{B}), \pi(x) \models \neg\chi$ or $\mathfrak{M}(\mathcal{B}), \pi(x) \models \neg\theta$, so by definition of $\mathfrak{M}(\mathcal{B})$ we derive $\mathfrak{M}(\mathcal{B}), \pi(x) \models \neg(\chi \wedge \theta)$. The latter can be proved similarly.

$\psi = \Diamond\chi$. We have that $\nu(\neg\Diamond\chi, x) \in \mathcal{B}$. Let $\pi(y)$ be arbitrary element of $\mathbf{W}$ such that $(\pi(x), \pi(y)) \in \mathbf{R}$. By construction of $\mathbf{R}$ there are terms u, v such that $R(u, v) \in \mathcal{B}, x \approx u \in \mathcal{B}, y \approx v \in \mathcal{B}$. Since $\mathcal{B}$ is open and fully expanded, by the equality rules, we have that $R(x, y) \in \mathcal{B}$. Thus, by the rule $(\neg\Diamond)$, we obtain $\nu(\neg\chi, y)$. By induction hypothesis we have $\mathfrak{M}(\mathcal{B}), \pi(y) \models \neg\chi$. Because $\pi(y)$ was arbitrarily chosen, we have $\mathfrak{M}(\mathcal{B}), \pi(x) \models \neg\Diamond\chi$.

$\psi = \mathsf{E}_{>n}\chi$. Let $\pi(x_1), \ldots, \pi(x_{n+1}) \in \mathbf{W}$ and such that $\pi(x_i) \neq \pi(x_j)$ for $0 < i < j \leq n + 1$. It follows that $x_i \approx x_i \in \mathcal{B}$ for $i = 1, \ldots, n + 1$ and $x_i \not\approx x_j \in \mathcal{B}$ for $0 < i < j \leq n + 1$. Since the rule $(\neg\mathsf{E}_{>n})$ is applied to $\nu(\neg\mathsf{E}_{>n}\chi, x)$ in $\mathcal{B}$ we have $\nu(\neg\chi, x_i) \in \mathcal{B}$ for some $i = 1, \ldots, n + 1$. By inductive hypothesis we get $\mathfrak{M}(\mathcal{B}), \pi(x_i) \models \neg\chi$. Since $n + 1$-tuple of elements of $\mathbf{W}$ was picked arbitrarily, by definition of $\mathfrak{M}(\mathcal{B})$ we obtain $\mathfrak{M}(\mathcal{B}), \pi(x) \models \neg\mathsf{E}_{>n}\chi$.

$\varphi = \psi \wedge \chi$. By application of $(\wedge)$ to $\nu(\psi \wedge \chi, x)$, we have $\nu(\psi, x) \in \mathcal{B}$ and $\nu(\chi, x) \in \mathcal{B}$. By inductive hypothesis we have $\mathfrak{M}(\mathcal{B}), \pi(x) \models \psi$ and $\mathfrak{M}(\mathcal{B}), \pi(x) \models \chi$ and, thus, $\mathfrak{M}(\mathcal{B}), \pi(x) \models \psi \wedge \chi$.

$\varphi = \Diamond\psi$. Since the rule $(\Diamond)$ is applied to $\nu(\Diamond\psi, x)$ in $\mathcal{B}$ we have $R(x, f(\psi, x) \in \mathcal{B}$ and $\nu(\psi, f(\psi, x) \in \mathcal{B}$. By construction of $\mathbf{R}$ we conclude that $(\pi(x), \pi(f(\psi, x))) \in \mathbf{R}$. Furthermore, by induction hypothesis, $\mathfrak{M}(\mathcal{B}), \pi(f(\psi, x)) \models \psi$. Hence, $\mathfrak{M}(\mathcal{B}), \pi(x) \models \Diamond\psi$ by definition of truth relation.

$\varphi = \mathsf{E}_{>n}\psi$. Because $\mathcal{B}$ is open and fully expanded, so $(\mathsf{E}_{>n})$-rule has been applied to $\mathsf{E}_{>n}\psi$. Therefore, it follows that $\nu(\psi, x_1) \in \mathcal{B}, \ldots, \nu(\psi, x_{n+1}) \in \mathcal{B}$ and $x_i \not\approx x_j \in \mathcal{B}$ for $0 < i < j \leq n + 1$. By induction hypothesis and by construction of $\mathbf{W}$ we obtain that $\mathfrak{M}(\mathcal{B}), \pi(x_i) \models \psi$ for $i = 1, \ldots, n+1$. Since $\mathcal{B}$ is open $x_i \approx x_j \notin \mathcal{B}$ for $0 < i < j \leq n + 1$ and, hence, $\pi(x_i) \neq \pi(x_j)$ for $0 < i < j \leq n+1$. By definition of truth relation this means $\mathfrak{M}(\mathcal{B}) \models \mathsf{E}_{>n}\psi$. □

Theorem A.2 $\mathcal{T}_{\mathbf{K}(\mathsf{E}_n)}$ *is a constructively complete tableau calculus for* $\mathbf{K}(\mathsf{E}_n)$.

Proof Follows from the construction of $\mathfrak{M}(\mathcal{B})$ and Lemma A.1. $\square$

B Proofs of other propositions and theorems

Proof of Proposition 5.1. Henceforth, without loss of generality, we treat Γ as $\bigwedge \Gamma = \bigwedge\{\psi \mid \psi \in \Gamma\}$. We search for a branch $\mathcal{B}$ and $\mathfrak{M}(\mathcal{B}) = \langle \mathbf{W}, \mathbf{R}, \mathbf{V} \rangle$ such that $\mathsf{Card}(\mathbf{W}) \leq \mathsf{Card}(\mathbf{U})$. We construct $\mathcal{B}$ inductively on application of the rules in derivation.

Base case: $\varphi = \Gamma$**.** We have $\nu(\Gamma, x)$ in an initial step of our derivation. Since Γ is satisfied by $\mathfrak{N}$, there exists $\mathbf{x} \in \mathbf{U}$ such that $\mathfrak{N}, \mathbf{x} \models \Gamma$. We set $x = \mathbf{x}$.

Inductive step: We assume that up to the n-th step of derivation we matched each expression of the domain sort, that have occurred in the tableau, with an appropriate world in $\mathfrak{N}$

($\wedge$) Assume that in the n-th step of derivation we obtained a formula of the form $\nu(\varphi \wedge \psi, x)$. By the inductive hypothesis, there exists $\mathbf{x} \in \mathbf{U}$ such that $\mathfrak{N}, \mathbf{x} \models \varphi \wedge \psi$. By definition of a model, it follows that $\mathfrak{N}, \mathbf{x} \models \varphi$ and $\mathfrak{N}, \mathbf{x} \models \psi$, so after applying ($\wedge$) and obtaining $\nu(\varphi, x)$ and $\nu(\psi, x)$, matching $x = \mathbf{x}$ still holds.

($\neg\wedge$) We have a formula of the form $\neg\nu(\varphi \wedge \psi, x)$. By the inductive hypothesis, there exists $\mathbf{x} \in \mathbf{U}$ such that $\mathfrak{N}, \mathbf{x} \models \neg(\varphi \wedge \psi)$. By definition of a model, it follows that $\mathfrak{N}, \mathbf{x} \not\models \varphi$ or $\mathfrak{N}, \mathbf{x} \not\models \psi$. After applying ($\neg\wedge$) which is a branching rule, we obtain $\neg\nu(\varphi, x)$ in the left branch and $\neg\nu(\psi, x)$ in the second branch. We choose left branch in the first case and right branch in the second and leave the matching $\mathbf{x} = x$.

($\Diamond$) In the n-th step of derivation we have a formula of the form $\nu(\Diamond\varphi, x)$. By the inductive hypothesis, there exists $\mathbf{x} \in \mathbf{U}$ such that $\mathfrak{N}, \mathbf{x} \models \Diamond\varphi$ and, therefore, that there is $\mathbf{y} \in \mathbf{U}$ such that $\mathbf{S}(\mathbf{x}, \mathbf{y})$ holds and $\mathfrak{N}, \mathbf{x} \models \varphi$. After applying ($\Diamond$) we obtain $R(x, f(\Diamond\varphi, x))$ and $\nu(\varphi, f(\Diamond\varphi, x))$. We set $\mathbf{y} = f(\Diamond\varphi, x)$.

($\neg\Diamond$) We have formulas of the form $\neg\nu(\Diamond\varphi, x)$ and $y \approx y$. Since y must have occurred in $\mathcal{B}$ so far, we have already fixed a particular $\mathbf{y} \in \mathbf{U}$ that $\mathbf{y} = y$. By the inductive hypothesis, there exists $\mathbf{x} \in \mathbf{U}$ such that $\mathfrak{N}, \mathbf{x} \models \neg\Diamond\varphi$ and, therefore, that either $\mathbf{S}(\mathbf{x}, \mathbf{y})$ does not hold or $\mathfrak{N}, \mathbf{x} \not\models \varphi$. After applying ($\neg\Diamond$) being a branching rule we obtain $\neg R(x, y)$ and $\neg\nu(\varphi, y)$. If the former is the case we choose the left branch, if the latter is the case we choose the right branch and leave the matching $\mathbf{x} = x$, $\mathbf{y} = y$.

($\mathsf{E}_{>n}$) In the n-th step of derivation we have a formula of the form $\nu(\mathsf{E}_{>n}\varphi, x)$. By the inductive hypothesis, there exists $\mathbf{x} \in \mathbf{U}$ such that $\mathfrak{N}, \mathbf{x} \models \mathsf{E}_{>n}\varphi$ and, therefore, that there is $n+1$-tuple of distinct worlds $\mathbf{x_1}, \ldots, \mathbf{x_{n+1}} \in \mathbf{U}$ such that $\mathfrak{N}, \mathbf{x_1} \models \varphi \ldots \mathfrak{N}, \mathbf{x_{n+1}} \models \varphi$. After applying ($\mathsf{E}_{>n}$) we obtain and $\nu(\varphi, f_1(\mathsf{E}_{>n}\varphi, x)), \ldots, \nu_{n+1}(\varphi, f_1(\mathsf{E}_{>n}\varphi, x))$ and $x_i \not\approx x_j$ for $0 < i < j \leq n+1$. We set $\mathbf{x_1} = f_1(\mathsf{E}_{>n}\varphi, x), \ldots, \mathbf{x_{n+1}} = f_{n+1}(\mathsf{E}_{>n}\varphi, x)$.

($\neg\mathsf{E}_{>n}$) We have formulas of the form $\neg\nu(\mathsf{E}_{>n}\varphi, x), y_1 \approx y_1, \ldots, y_{n+1} \approx y_{n+1}$.

$y_1, \ldots, y_{n+1}$ must have already occurred in $\mathcal{B}$, so we matched them with particular $\mathbf{y_1}, \ldots, \mathbf{y_{n+1}}$. By the inductive hypothesis, there exists $\mathbf{x} \in \mathbf{U}$ such that $\mathfrak{N}, \mathbf{x} \not\models \mathsf{E}_{>n}\varphi$. It means that either $\mathfrak{N}, \mathbf{y_i} \not\models \varphi$ for some $i \in \{1, \ldots, n+1\}$ or $\mathbf{y_i} \neq \mathbf{y_j}$ for some $0 < i < j \leq n+1$. After application of $(\neg \mathsf{E}_{>n})$ we pick the correct branch, subject to the case that turned out to hold for $\mathfrak{N}$.

(ub) Two formulas $x \approx x$ and $y \approx y$ have already occurred in the branch. It means that we affixed two worlds $\mathbf{x}, \mathbf{y} \in \mathbf{U}$ to them, respectively. It either is the case that $\mathbf{x} = \mathbf{y}$ or $\mathbf{x} \neq \mathbf{y}$. After applying (ub) we chose left branch if the former is the case and we chose the right branch if the latter holds.

Thus, we showed that if there is a Γ-model $\mathfrak{N} = \langle \mathbf{U}, \mathbf{S}, \mathbf{Z} \rangle$ then in a tableau derivation we can find a branch $\mathcal{B}$ and a model $\mathfrak{M}(\mathcal{B}) = \langle \mathbf{W}, \mathbf{R}, \mathbf{V} \rangle$, and subsequently fix a function $g : \mathbf{U} \longrightarrow \mathbf{W}$ defined as follows:

$$g(\mathbf{x}) = \begin{cases} [x]_{\approx}, & \text{if there is } x \approx x \in \mathcal{B} \text{ such that } x \text{ was affixed} \\ & \text{to } \mathbf{x} \text{ whilst derivation} \\ \text{arbitrary element of } \mathbf{W}, & \text{otherwise} \end{cases}$$

By construction of $\mathfrak{M}(\mathcal{B})$ it follows that g is onto. Hence the conclusion.

Proof of Theorem 5.2. Let φ be a formula satisfiable on a (possibly infinite) model $\mathfrak{M} = \langle W, R, V \rangle$. We show that there exists a finite model $\mathfrak{M}' = \langle W', R', V' \rangle$ on which φ is satisfiable.

We proceed in two steps.

In the first step we exploit a filtration-like method to divide the universe W into finite number of equivalence classes. We fix the equivalence relation $\leftrightsquigarrow_{\mathrm{Sub}(\varphi)}$ in the following way:

$$w \leftrightsquigarrow_{\mathrm{Sub}(\varphi)} v \text{ iff for all } \psi \in \mathrm{Sub}(\varphi)\ (\mathfrak{M}, w \models \psi \text{ iff } \mathfrak{M}, v \models \psi).$$

It is straightforward that there are only finitely many such equivalence classes, namely $2^{\mathsf{Card}(\mathrm{Sub}(\varphi))}$.

In the second step we abandon the ordinary filtration procedure. Instead of merging all worlds from the equivalence classes, we reduce the cardinality of each class in the following manner. Let $[w] \subseteq W$ be an arbitrary $\leftrightsquigarrow_{\mathrm{Sub}(\varphi)}$-equivalence class. If $\mathsf{Card}([w]) > n+1$ then we delete all but $n+1$ arbitrary worlds from $[w]$. If $\mathsf{Card}([w]) \leq n+1$ then we leave $[w]$ unchanged. Next, from each reduced equivalence class $\mathbf{w'}$ we pick an arbitrary representative w_0. We set a new model $\mathfrak{M}' = \langle W', R', V' \rangle$ where $W = \bigcup_{[w] \in W/\leftrightsquigarrow_{\mathrm{Sub}(\varphi)}} [w']$, $R' = R \downharpoonleft W' \cup \bigcup_{[w],[v] \in W/\leftrightsquigarrow_{\mathrm{Sub}(\varphi)}} \{(w, v_0) \mid w \in [w'], (w, v) \in R, v \in \mathbf{v} \setminus [v']\}$ and $V' = V \downharpoonleft W'$.

We prove that $\mathfrak{M}'$ is a model for φ by induction on the complexity of the elements of $\mathrm{Sub}(\varphi)$.

The Boolean cases are obvious and follow directly from the definition of W' and V'.

Suppose that a formula $\Diamond\psi$ is satisfiable on $\mathfrak{M}$. It means that there exists such $w \in W$ that $\mathfrak{M}, w \models \Diamond\psi$. We pick arbitrary $w' \in [w']$. By definition of

$\leftrightsquigarrow_{\mathrm{Sub}(\varphi)}$ it follows that $\mathfrak{M}, w' \models \Diamond\psi$. Consequently, we can find $v \in W$ such that $(w', v) \in R$ and $\mathfrak{M}, v \models \psi$. If $v \in [v']$ then we have also $\mathfrak{M}', w' \models \Diamond\psi$. Otherwise, by definition of $\leftrightsquigarrow_{\mathrm{Sub}(\varphi)}$ and R' there exists $v_0 \in [v']$ such that $(w', v_0) \in R'$ and $\mathfrak{M}, v_0 \models \psi$. Therefore, $\Diamond\psi$ is satisfiable on $\mathfrak{M}$.
Now, assume that a formula $\mathsf{E}_{>m}\psi$ is satisfied by $\mathfrak{M}$. It means that there exist more than m worlds in which ψ holds. Two cases may occur. Either ψ holds in elements of (at least one) equivalence class $[w]$ such that $\mathsf{Card}([w]) > n$. Then, by the construction of $\mathbf{w}'$, we obtain that $\mathsf{E}_{>m}$ is satisfied by $\mathfrak{M}'$. Otherwise ψ holds in elements of equivalence classes $[w]_{i_1}, \ldots, [w]_{i_k}$ such that $\mathsf{Card}([w]_{i_j} = \leq n$ and $\sum_{j=1}^{k} \mathsf{Card}([w]_{i_j}) > n$. But by construction of $[w]'_{i_j}$ these classes remained unchanged in W', therefore $\sum_{j=1}^{k} \mathsf{Card}([w']_{i_j}) > n$. It follows that $\mathsf{E}_{>m}$ is satisfied by $\mathfrak{M}'$. Obviously, reduction of the size of W cannot disturb satisfiability of the formulas $\mathsf{E}_{<m}\psi$ on $\mathfrak{M}'$.
Since $\mathsf{Card}(W/\leftrightsquigarrow_{\mathrm{Sub}(\varphi)}) = 2^{\mathrm{Sub}(\varphi)}$ and for each $[w']$ obtained from $[w] \in W/\leftrightsquigarrow_{\mathrm{Sub}(\varphi)}$ $\mathsf{Card}([w']) \leq n+1$, it is clear that $\mathsf{Card}(W') \leq 2^{\mathsf{Card}(\{\mathrm{Sub}(\varphi)\})+\log(n+1)}$. That completes the proof.

Proof of Proposition 6.3. The case of $(\mathrm{Tr})^R$ is trivial. It follows from construction of $\mathbf{W}$ and transitivity of identity relation between worlds in $\mathbf{W}$. The case of $(\neg\Diamond)^R$ can be proven in the following way. Let φ be an arbitrary formula occurred in $\mathcal{B}$ and reflected in $\mathfrak{M}(\mathcal{B})$. Assume $\neg\nu(\Diamond\varphi, x) \in \mathcal{B}$ and a domain term y is occurred in $\mathcal{B}$. We must prove that either $R(\pi(x), \pi(y))$ does not hold in $\mathfrak{M}(\mathcal{B})$ or $\mathfrak{M}(\mathcal{B}), \pi(y) \not\models \varphi$. Suppose $\mathbf{R}(\pi(x), \pi(y))$ holds. By construction of $\mathfrak{M}(\mathcal{B})$ it means that $R(z, v), z \approx x, v \approx y$ must have occurred in $\mathcal{B}$. Thus, by the equality rules $R(x, y) \in \mathcal{B}$ and, consequently, the rule $(\neg\Diamond)^R$ has been applied in $\mathcal{B}$. Therefore, $\neg\nu(\varphi, y) \in \mathcal{B}$. Because φ is reflected in $\mathfrak{M}(\mathcal{B})$ this means that $\mathfrak{M}(\mathcal{B}).\pi(y) \not\models \varphi$.

Proof of Proposition 7.1. Case: $\nu(\varphi, x), \neg\nu(\varphi, x)$. $(\leftarrow)$ Suppose that a formula φ holds in world x (which is expressed by $\nu(\varphi, x)$). Then we pick a conservative extension of $\mathfrak{M}$ by introducing a fresh propositional variable p_x and extending v to v' in such a way that v' agrees with v on all propositional variables from Γ and $v'(p_x) = \{x\}$. Then, by definition of $\mathsf{E}_{=1}$, $\mathsf{E}_{=1}p_x$ holds and since p_x holds a unique world, namely x where also φ holds, for no world it is the case that both p_x and $\neg\varphi$ hold. Therefore, $\mathsf{E}_{=1}p_x \wedge \mathsf{E}_{=0}(p_x \wedge \neg\varphi)$ is satisfied on $\mathfrak{M}'$.

$(\rightarrow)$ Suppose that a formula $\mathsf{E}_{=1}p_x \wedge \mathsf{E}_{=0}(p_x \wedge \neg\varphi)$ is satisfied on $\mathfrak{M}'$ is satisfied on a conservative extension $\mathfrak{M}' = \langle W, R, V' \rangle$, where $v'(p_x) = \{x\}$. It therefore follows that for no world both p_x and $\neg\varphi$ hold. Since x is the only world where p_i holds, it must be the case that φ holds in x. The case of $\neg\nu(\varphi, x)$ proceeds analogically.

Case: $R(x, y), \neg R(x, y)$. $(\leftarrow)$ Suppose that $R(x, y)$ holds. We pick a conservative extension of $\mathfrak{M}$ by introducing two fresh propositional variables p_x and p_y and extending v to v' such that they agree on all propositional variables from Γ and, additionally, $v'(p_x) = \{x\}$ and $v'(p_y) = \{y\}$. Since p_y holds in y and $(x, y) \in R$, then by (1) we obtain that $\Diamond p_y$ holds in x. x is the unique world where p_x holds so for all worlds it cannot be the case that $p_x \wedge \neg\Diamond p_y$ holds. Hence the conclusion.

($\rightarrow$) Assume that $\mathsf{E}_{=1}p_x \wedge \mathsf{E}_{=1}p_y \wedge \mathsf{E}_{=0}(p_x \wedge \neg\Diamond p_y)$ is satisfied in a conservative extension of $\mathfrak{M}$ where $v'(p_x) = \{x\}$ and $v'(p_y) = \{y\}$. For no world in which p_x is true $\neg\Diamond p_y$ is also true. It follows that $\Diamond p_y$ holds in x and, by (1), $R(x, y)$. We proceed similarly for $\neg R(x, y)$.

Case: $x \approx y, x \not\approx y$. ($\leftarrow$) Suppose that $x \approx y$ holds. It means that worlds x and y coincide on $\mathfrak{M}$. We pick a conservative extension of $\mathfrak{M}$ by adding two fresh propositional variables p_x and p_y and extending v to v' in such way that $v'(p_x) = \{x\}$ and $v'(p_y) = \{y\}$. Since x and y are the same world which is the unique world where either, p_x and p_y hold, it means that there is no world in which $p_x \wedge \neg p_y$ hold. It completes this part of the proof.

($\rightarrow$) Suppose that $\mathsf{E}_{=1}p_x \wedge \mathsf{E}_{=1}p_y \wedge \mathsf{E}_{=0}(p_x \wedge \neg p_y)$ is satisfied on $\mathfrak{M}'$, a conservative extension of $\mathfrak{M}$ where p_x, p_y are fresh and $v'(p_x) = \{x\}, v'(p_y) = \{y\}$. Since there is no world where both p_x and $\neg p_y$ hold, it must be the case that in all worlds in which p_x holds, p_y also holds. But there is only one world satisfying p_x, namely x, and one world satisfying y, namely y. Since conjunction of both these hold in both worlds, by (1) they must coincide. We conduct a similar proof for $x \not\approx y$.