Clausal Resolution for Modal Logics of Confluence – Extended Version

João Marcos[1], Cláudia Nalon[2*], and Clare Dixon[3]

[1] LoLITA and Dept. of Informatics and Applied Mathematics, UFRN, Brazil
jmarcos@dimap.ufrn.br
[2] Departamento de Ciência da Computação, Universidade de Brasília
Caixa Postal 4466 – CEP:70.910-090 – Brasília – DF – Brazil
nalon@unb.br
[3] Department of Computer Science, University of Liverpool
Liverpool, L69 3BX – United Kingdom
CLDixon@liverpool.ac.uk

Abstract. We present a clausal resolution-based method for normal modal logics of confluence, whose Kripke semantics are based on frames characterized by appropriate instances of the Church-Rosser property. Differently from other approaches, where inference rules are based on the syntax of a particular set of axioms, we focus on the restrictions imposed on the binary accessibility relation for each particular normal modal logic.

Keywords: normal modal logics, combined logics, resolution method

1 Introduction

Modal logics are often introduced as extensions of classical logic with two additional unary operators: '$\Box$' and '$\Diamond$', whose meanings depend on the framework in which they are defined. In the most common interpretation, formulae '$\Box p$' and '$\Diamond p$' are to be read as "p is necessary" and "p is possible", respectively. Evaluation of a modal formula depends on a set of interpretations, also called set of *possible worlds*. Given a world w, the set of worlds which are *accessible* from w, is defined. Different modal logics assume different *accessibility relations* between worlds. Worlds and their accessibility relations define a structure known as a *Kripke structure*. The evaluation of a formula depends on this structure. Given an appropriate accessibility relation and a world w, a formula $\Box p$ is true at w if p is true at all worlds accessible from w; $\Diamond p$ is true at w if p is true at some world accessible from w.

In normal modal logics, the schema $\Box(\varphi \Rightarrow \psi) \Rightarrow (\Box\varphi \Rightarrow \Box\psi)$ (the axiom **K**), where φ and ψ are well-formed formulae and $\Rightarrow$ stands for classical

* C. Nalon was supported by CAPES Foundation BEX 8712/11-5.

implication, is valid. The weakest of these logics, where the axioms for classical propositional logic and the axiom **K** hold, is named $\mathsf{K}_{(1)}$. They correspond to the class of Kripke structures with no restrictions imposed on the accessibility relation. In the multi-modal version, named $\mathsf{K}_{(n)}$, Kripke structures are directed multigraphs and modal operators are equipped with indexes over a set of *agents*, given by $\mathcal{A} = \{1, 2, \ldots, n\}$, for some positive integer n. Accordingly, in this case classical logic is extended with operators $\boxed{1}, \boxed{2}, \ldots, \boxed{n}$, where a formula as $\boxed{a}p$, $a \in \mathcal{A}$, is read as "agent a considers p to be necessary". The modal operator $\Diamond_a$ is the dual of $\boxed{a}$ and is introduced as an abbreviation for $\neg\boxed{a}\neg$, where $\neg$ stands for classical negation. The logic $\mathsf{K}_{(n)}$ can be seen as the *fusion* of n copies of $\mathsf{K}_{(1)}$ and the axiomatisation is given by the union of the axioms for classical propositional logic with the axiomatic schemas $\boxed{a}(\varphi \Rightarrow \psi) \Rightarrow (\boxed{a}\varphi \Rightarrow \boxed{a}\psi)$, for each $a \in \mathcal{A}$.

A logic of confluence $\mathsf{K}_{(n)}^{p,q,r,s}$ is a modal system axiomatised by $\mathsf{K}_{(n)}$ plus axioms $\mathbf{G}_a^{p,q,r,s}$ of the form

$$\Diamond_a^{p}\boxed{a}^{q}\varphi \Rightarrow \boxed{a}^{r}\Diamond_a^{s}\varphi$$

where $a \leq n$, φ is a well-formed formula, $p, q, r, s \in \mathbb{N}$, where $\boxed{a}^0\varphi \stackrel{\text{def}}{=} \varphi$ and $\boxed{a}^{i+1}\varphi \stackrel{\text{def}}{=} \boxed{a}(\boxed{a}^{i}\varphi)$, where $\Diamond_a^0\varphi \stackrel{\text{def}}{=} \varphi$ and $\Diamond_a^{i+1}\varphi \stackrel{\text{def}}{=} \Diamond_a\Diamond_a^{i}\varphi$, for $i \in \mathbb{N}$ (the superscript is often omitted if equal to 1). Such axiomatic schemas are known as Scott-Lemmon axioms [**?**]. Using Modal Correspondence Theory, it can be shown that the frame condition on a logic where an instance of $\mathbf{G}_a^{p,q,r,s}$ is valid corresponds to a diamond-like structure generalizing the so-called Church-Rosser property, as illustrated in Fig. 1 [**?**]. To be a bit more precise, let $\mathcal{W}$ be a nonempty set of worlds and let $\mathcal{R}_a \subseteq \mathcal{W} \times \mathcal{W}$ be the accessibility relation of agent $a \in \mathcal{A}$. By $w\mathcal{R}_a^0 w'$ we shall mean that $w = w'$, and $w\mathcal{R}_a^{i+1}w'$ will denote that there is some world w'' such that $w\mathcal{R}_a w''$ and $w''\mathcal{R}_a^i w'$ (thus, $w\mathcal{R}_a^j w'$ holds good if there is a j-long $\mathcal{R}_a$-path from w to w'; alternatively, for that we also write $(w, w') \in \mathcal{R}_a^j$). From that perspective, the condition on frames that corresponds to the axiom $\mathbf{G}_a^{p,q,r,s}$ is described by $\forall w_0, w_1, w_2\ (w_0\mathcal{R}_a^p w_1 \wedge w_0\mathcal{R}_a^r w_2 \Rightarrow \exists w_3(w_1\mathcal{R}_a^q w_3 \wedge w_2\mathcal{R}_a^s w_3))$, where w_0, w_1, w_2, $w_3 \in \mathcal{W}$.

Many well-known modal axiomatic systems may be identified with particular logics of confluence. For instance, $\mathsf{T}_{(n)}$ corresponds to $\mathsf{K}_{(n)}^{0,1,0,0}$, i.e. where the axiom $\boxed{a}\varphi \Rightarrow \varphi$ is valid, for all $a \in \mathcal{A}$ and any formula φ. The axiom **4** can be written as $\mathbf{G}_a^{0,1,2,0}$, that is, $\boxed{a}^1\varphi \Rightarrow \boxed{a}^2\varphi$. The Geach axiom **G1**, given by $\mathbf{G}_a^{1,1,1,1}$ ($\Diamond_a\boxed{a}\varphi \Rightarrow \boxed{a}\Diamond_a\varphi$), is the converse of the so-called McKinsey's axiom. Formulae in $\mathsf{K}_{(n)}^{1,1,1,1}$ are satisfiable if, and only if, they are satisfiable in a *confluent* model.

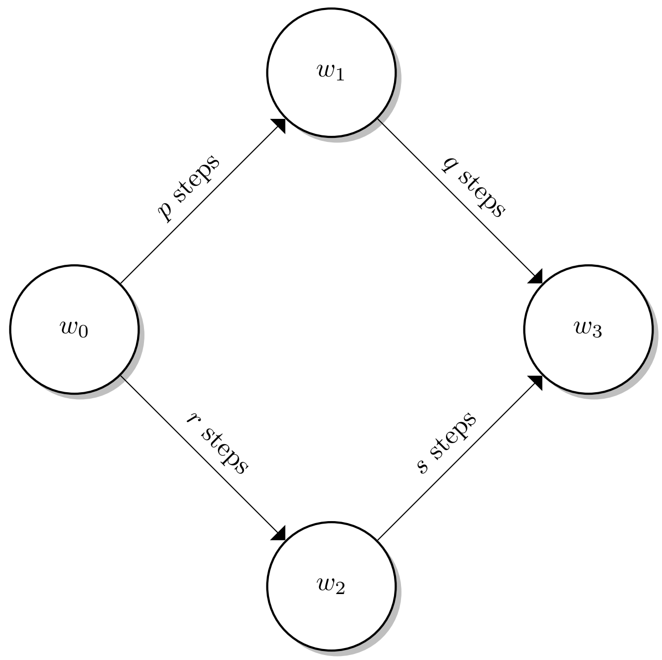

Fig. 1. The diamond property in frames where $\mathbf{G}_a^{p,q,r,s} = \langle a \rangle^p \boxed{a}^q \varphi \Rightarrow \boxed{a}^r \langle a \rangle^s \varphi$ is valid.

In this work, we restrict attention to logics where $p, q, r, s \in \{0, 1\}$. Table 1 shows the relevant axiomatic schemas, their standard names, and respective conditions on frames. Note that $\mathbf{G}_a^{0,0,0,0}$, $\mathbf{G}_a^{0,1,1,0}$, and $\mathbf{G}_a^{1,0,0,1}$ are instances of classical tautologies and are not included in Table 1. Also, given the duality between $\boxed{a}$ and $\langle a \rangle$, $\mathbf{G}_a^{p,q,r,s}$ is semantically equivalent to $\mathbf{G}_a^{r,s,p,q}$. Thus, there are in fact eight families of multi-modal logics related to the axioms $\mathbf{G}_a^{p,q,r,s}$, where $p, q, r, s \in \{0, 1\}$.

We present a clausal resolution-based method for solving the satisfiability problem in logics axiomatised by $\mathbf{K}$ plus $\mathbf{G}_a^{p,q,r,s}$, where $p, q, r, s \in \{0, 1\}$. The resolution calculus is based on that of [**?**], which deals with the logical fragment corresponding to $\mathsf{K}_{(n)}$. The new inference rules to deal with axioms of the form $\mathbf{G}_a^{p,q,r,s}$ add relevant information to the set of clauses: the conclusion of each inference rule ensures that properties related to the corresponding conditions on frames hold, that is, the newly added clauses capture the required properties of a model. We show that the proof method hereby presented is sound, complete, and terminating.

(p,q,r,s)	Name	Axioms	Property	Condition on Frames
$(0,0,1,1)$ $(1,1,0,0)$	**B**	$\varphi \Rightarrow \boxed{a}\Diamond_a\varphi$ $\Diamond_a\boxed{a}\varphi \Rightarrow \varphi$	symmetric	$\forall w, w'(w\mathcal{R}_a w' \Rightarrow w'\mathcal{R}_a w)$
$(0,0,1,0)$ $(1,0,0,0)$	**Ban**	$\varphi \Rightarrow \boxed{a}\varphi$ $\Diamond_a\varphi \Rightarrow \varphi$	modally banal	$\forall w, w'(w\mathcal{R}_a w' \Rightarrow w = w')$
$(0,1,0,1)$	**D**	$\boxed{a}\varphi \Rightarrow \Diamond_a\varphi$	serial	$\forall w \exists w'(w\mathcal{R}_a w')$
$(1,0,1,0)$	**F**	$\Diamond_a\varphi \Rightarrow \boxed{a}\varphi$	functional	$\forall w, w', w''((w\mathcal{R}_a w' \wedge w\mathcal{R}_a w'') \Rightarrow w' = w'')$
$(0,0,0,1)$ $(0,1,0,0)$	**T**	$\varphi \Rightarrow \Diamond_a\varphi$ $\boxed{a}\varphi \Rightarrow \varphi$	reflexive	$\forall w(w\mathcal{R}_a w)$
$(1,0,1,1)$ $(1,1,1,0)$	**5**	$\Diamond_a\varphi \Rightarrow \boxed{a}\Diamond_a\varphi$ $\Diamond_a\boxed{a}\varphi \Rightarrow \boxed{a}\varphi$	euclidean	$\forall w, w', w''((w\mathcal{R}_a w' \wedge w\mathcal{R}_a w'') \Rightarrow w'\mathcal{R}_a w'')$
$(1,1,1,1)$	**G1**	$\Diamond_a\boxed{a}\varphi \Rightarrow \boxed{a}\Diamond_a\varphi$	convergent	$\forall w, w', w''((w\mathcal{R}_a w' \wedge w\mathcal{R}_a w'') \Rightarrow \exists w'''(w'\mathcal{R}_a w''' \wedge w''\mathcal{R}_a w'''))$
$(0,1,1,1)$ $(1,1,0,1)$	$\mathbf{5}^{-1}$	$\boxed{a}\varphi \Rightarrow \boxed{a}\Diamond_a\varphi$ $\Diamond_a\boxed{a}\varphi \Rightarrow \Diamond_a\varphi$	converse of euclidean	$\forall w, w'(w\mathcal{R}_a w' \Rightarrow \exists w''(w\mathcal{R}_a w'' \wedge w'\mathcal{R}_a w''))$

Table 1. Conditions on Frames

2 The Normal Logic $\mathsf{K}_{(n)}$

As it was already mentioned, the weakest of the normal modal systems, known as $\mathsf{K}_{(n)}$, is an extension of the classical propositional logic with the operators $\boxed{a}$, $1 \leq a \leq n$, where the axiom $\mathbf{K}_a$, i.e. $\boxed{a}(\varphi \Rightarrow \psi) \Rightarrow (\boxed{a}\varphi \Rightarrow \boxed{a}\psi)$, holds. There is no restriction on the accessibility relation over worlds. As the subscript in $\mathsf{K}_{(n)}$ indicates, we consider the multi-agent version, that is, the fusion of several copies of $\mathsf{K}_{(1)}$.

Formulae are constructed from a denumerable set of *propositional symbols*, $\mathcal{P} = \{p, q, p', q', p_1, q_1, \ldots\}$. The finite set of agents is defined as $\mathcal{A} = \{1, \ldots, n\}$. Besides the usual propositional connectives ($\neg, \wedge, \vee, \Rightarrow$), we introduce a set of unary modal operators $\boxed{1}, \ldots, \boxed{n}$, where $\boxed{a}\varphi$ is read as "agent a considers φ necessary". When $n = 1$, we generally may omit the index, that is, we let $\Box\varphi$ stand for $\boxed{1}\varphi$. The fact that agent a considers φ to be possible, i.e. $\Diamond_a\varphi$ holds good, is denoted by asserting that $\neg\boxed{a}\neg\varphi$ holds good. We define next the language of $\mathsf{K}_{(n)}$ and of all other systems presented in this paper:

Definition 1. *The* set of well-formed formulae*,* $\mathsf{WFF}_{\mathsf{K}_{(n)}}$*, is the least set such that:*

- *the propositional symbols are in* $\mathsf{WFF}_{\mathsf{K}_{(n)}}$*;*
- **true** *is in* $\mathsf{WFF}_{\mathsf{K}_{(n)}}$*;*
- *if* φ *and* ψ *are in* $\mathsf{WFF}_{\mathsf{K}_{(n)}}$*, then so are* $\neg\varphi$*,* $(\varphi \wedge \psi)$*, and* $\boxed{a}\varphi$*, for each* $a \in \mathcal{A}$*.*

A *literal* is either a proposition or its negation; the set of literals is denoted by $\mathcal{L}$. By $\neg l$ we will denote the *complement* of the literal $l \in \mathcal{L}$, that is, $\neg l$ shall denote $\neg p$ if l is the propositional symbol p, and $\neg l$ shall denote p if l is the literal $\neg p$. A *modal literal* is either $\boxed{a}l$ or $\neg\boxed{a}l$, where $l \in \mathcal{L}$ and $a \in \mathcal{A}$.

The semantics of $\mathsf{K}_{(n)}$ is given, as usual, in terms of Kripke structures.

Definition 2. *By a* Kripke structure $\mathcal{S}$ for n agents over $\mathcal{P}$ *we mean a tuple* $(\mathcal{W}, w_0, \mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_n)$*, where* $\mathcal{W}$ *is a set of possible* worlds *(or* states*) with a distinguished world* w_0*, and each* $\mathcal{R}_a$ *is a binary relation on* $\mathcal{W}$*. A* Kripke model $\mathcal{M} = (\mathcal{S}, \pi)$ *equips a Kripke structure* $\mathcal{S}$ *with a function* $\pi : \mathcal{W} \to (\mathcal{P} \to \{true, false\})$ *that plays the role of an interpretation that associates to each state* $w \in \mathcal{W}$ *a truth-assignment to propositions.*

The binary relation $\mathcal{R}_a$ captures the accessibility relation according to agent a: a pair (w, w') is in $\mathcal{R}_a$ if agent a considers world w' possible, given her information in world w. We write $\langle \mathcal{M}, w \rangle \models \varphi$ to say that φ is true at world w in the Kripke model $\mathcal{M}$, and write $\langle \mathcal{M}, w \rangle \not\models \varphi$ to say that φ is false at w.

Definition 3. *Truth of a formula is defined as follows:*

- $\langle \mathcal{M}, w \rangle \models$ **true**
- $\langle \mathcal{M}, w \rangle \models p$ *if, and only if,* $\pi(w)(p) = true$*, where* $p \in \mathcal{P}$
- $\langle \mathcal{M}, w \rangle \models \neg\varphi$ *if, and only if,* $\langle \mathcal{M}, w \rangle \not\models \varphi$
- $\langle \mathcal{M}, w \rangle \models (\varphi \wedge \psi)$ *if, and only if,* $\langle \mathcal{M}, w \rangle \models \varphi$ *and* $\langle \mathcal{M}, w \rangle \models \psi$
- $\langle \mathcal{M}, w \rangle \models \boxed{a}\varphi$ *if, and only if* $\langle \mathcal{M}, w' \rangle \models \varphi$*, for all* w' *such that* $w \mathcal{R}_a w'$

The formulae **false**, $(\varphi \vee \psi)$, $(\varphi \Rightarrow \psi)$, and $\Diamond_a \varphi$ are introduced as the usual abbreviations for $\neg$**true**, $\neg(\neg\varphi \wedge \neg\psi)$, $(\neg\varphi \vee \psi)$, and $\neg\boxed{a}\neg\varphi$, respectively. Formulae are interpreted with respect to the distinguished world w_0, that is, satisfiability is defined with respect to pointed-models. Intuitively, w_0 is the world from which we start reasoning. A formula φ is said to be *satisfied in the model* $\mathcal{M} = (\mathcal{S}, \pi)$ *of the Kripke structure* $\mathcal{S} = (\mathcal{W}, w_0, \mathcal{R}_1, \ldots, \mathcal{R}_n)$ if $\langle \mathcal{M}, w_0 \rangle \models \varphi$; φ is *satisfiable in a Kripke structure* $\mathcal{S}$ if there is a model $\mathcal{M}$ of $\mathcal{S}$ such that $\langle \mathcal{M}, w_0 \rangle \models \varphi$; and φ is said to be *valid in a class* $\mathcal{E}$ *of Kripke structures* if it is satisfied in any model of any Kripke structure belonging to the class $\mathcal{E}$.

3 Resolution for $\mathsf{K}_{(n)}$

In [?], a sound, complete, and terminating resolution-based method for $\mathsf{K}_{(n)}$ is introduced. The approach is clausal: a formula to be tested for (un)satisfiability is firstly translated into a normal form, explained in Section 3.1, and then the inference rules given in Section 3.2 are applied until either a contradiction is found or no new clauses can be generated.

3.1 A Normal Form for $\mathsf{K}_{(n)}$

Formulae in the language of $\mathsf{K}_{(n)}$ can be transformed into a normal form called Separated Normal Form for Normal Logics (SNF_K). As the semantics is given with respect to a pointed-model, we add a nullary connective **start** in order to represent the world from which we start reasoning. Formally, given a model $\mathcal{M} = (\mathcal{W}, w_0, \pi, \mathcal{R}_1, \ldots, \mathcal{R}_n)$, we have that $\langle \mathcal{M}, w \rangle \models$ **start** if, and only if, $w = w_0$. A formula in SNF_K is represented by a conjunction of clauses, which are true at all reachable states, that is, they have the general form

$$\Box^* \bigwedge_i A_i$$

where A_i is a clause and $\Box^*$, the universal operator, characterized by the following truth-condition: $\langle \mathcal{M}, w \rangle \models \Box^* \varphi$ if, and only if, $\langle \mathcal{M}, w \rangle \models \varphi$, and for all w' such that $(w, w') \in \mathcal{R}_a$, for any $a \in \mathcal{A}$, $\langle \mathcal{M}, w' \rangle \models \Box^* \varphi$. Observe that φ must hold at the actual world w and at every world reachable from w, where reachability is defined in the usual way. The universal operator, which surrounds all clauses, ensures that the *translation* of a formula is true at all reachable worlds. Clauses are in one of the following forms:

- Initial clause $\qquad \textbf{start} \Rightarrow \bigvee_{b=1}^{r} l_b$
- Literal clause $\qquad \textbf{true} \Rightarrow \bigvee_{b=1}^{r} l_b$
- Positive a-clause $\qquad l' \Rightarrow \boxed{a}\, l$
- Negative a-clause $\qquad l' \Rightarrow \neg \boxed{a}\, l$

where $l, l', l_b \in \mathcal{L}$. Positive and negative a-clauses are together known as *modal a-clauses*; the index may be omitted if it is clear from the context.

The translation to SNF_K uses the renaming technique [**?**], where complex subformulae are replaced by new propositional symbols and the truth of these new symbols is linked to the formulae that they replaced in all states. We refer the reader to [**?**] for details on the transformation rules and their correctness. In the following, we show here, by means of an example, how the transformation is applied.

Example 1. Let φ be the formula $\neg(\boxed{1}(a \Rightarrow b) \Rightarrow (\boxed{1}a \Rightarrow \boxed{1}b))$. We show how to translate φ into its normal form. First we anchor φ to the initial state, that is, we have the clauses:

$$
\begin{array}{ll}
1. & \textbf{start} \Rightarrow t_1 \\
2. & \quad t_1 \Rightarrow \neg(\boxed{1}(a \Rightarrow b) \Rightarrow (\boxed{1}a \Rightarrow \boxed{1}b))
\end{array}
$$

where (1) is in the normal form, but clause (2) is not. As $\neg(\boxed{1}(a \Rightarrow b) \Rightarrow (\boxed{1}a \Rightarrow \boxed{1}b))$ is semantically equivalent to a conjunctive formula, (2) is rewritten as:

$$
\begin{array}{l}
3.\ t_1 \Rightarrow \boxed{1}(a \Rightarrow b) \\
4.\ t_1 \Rightarrow \neg(\boxed{1}a \Rightarrow \boxed{1}b))
\end{array}
$$

and, for the same reason, (4) is rewritten as:

$$
\begin{array}{l}
5.\ t_1 \Rightarrow \boxed{1}a \\
6.\ t_1 \Rightarrow \neg\boxed{1}b
\end{array}
$$

As clause (3) has a complex formula in the scope of $\boxed{1}$, we introduce a new propositional symbol, t_2, and rewrite (3) as:

$$
\begin{array}{l}
7.\ t_1 \Rightarrow \boxed{1}t_2 \\
8.\ t_2 \Rightarrow (a \Rightarrow b)
\end{array}
$$

Finally, (8) is rewritten into the normal form, that is, we obtain:

$$9.\ \textbf{true} \Rightarrow \neg t_2 \vee \neg a \vee b$$

The final set of clauses is given by:

$$
\begin{array}{ll}
1. & \textbf{start} \Rightarrow t_1 \\
5. & \quad t_1 \Rightarrow \boxed{1}\, a \\
6. & \quad t_1 \Rightarrow \neg \boxed{1}\, b \\
7. & \quad t_1 \Rightarrow \boxed{1}\, t_2 \\
9. & \textbf{true} \Rightarrow \neg t_2 \lor \neg a \lor b
\end{array}
$$

3.2 Inference Rules for $\mathsf{K}_{(n)}$

In the following, l, l', l_i, $l'_i \in \mathcal{L}$ ($i \in \mathbb{N}$) and D, D' are disjunctions of literals.

Literal Resolution. This is classical resolution applied to the propositional portion of the combined logic. An initial clause may be resolved with either a literal clause or an initial clause (**IRES1** and **IRES2**).

$$
[\textbf{IRES1}] \quad \frac{\begin{array}{l} \Box^*(\textbf{true} \Rightarrow D \lor l) \\ \Box^*(\textbf{start} \Rightarrow D' \lor \neg l) \end{array}}{\Box^*(\textbf{start} \Rightarrow D \lor D')}
\qquad
[\textbf{IRES2}] \quad \frac{\begin{array}{l} \Box^*(\textbf{start} \Rightarrow D \lor l) \\ \Box^*(\textbf{start} \Rightarrow D' \lor \neg l) \end{array}}{\Box^*(\textbf{start} \Rightarrow D \lor D')}
$$

Literal clauses may be resolved together (**LRES**):

$$
[\textbf{LRES}] \quad \frac{\begin{array}{l} \Box^*(\textbf{true} \Rightarrow D \lor l) \\ \Box^*(\textbf{true} \Rightarrow D' \lor \neg l) \end{array}}{\Box^*(\textbf{true} \Rightarrow D \lor D')}
$$

Modal Resolution. These rules are applied between clauses which refer to the same context, that is, they must refer to the same agent. For instance, we amy resolve two or more $\boxed{1}$-clauses (**MRES** and **GEN2**); or several $\boxed{1}$-clauses and a literal clause (**GEN1** and **GEN3**). The modal inference rules are (assuming m¿0):

[MRES]
$$\begin{array}{c} \Box^*(l_1 \Rightarrow \boxed{a}\, l) \\ \Box^*(l_2 \Rightarrow \neg \boxed{a}\, l) \\ \hline \Box^*(\mathbf{true} \Rightarrow \neg l_1 \vee \neg l_2) \end{array}$$

[GEN2]
$$\begin{array}{c} \Box^*(l'_1 \Rightarrow \boxed{a}\, l_1) \\ \Box^*(l'_2 \Rightarrow \boxed{a}\, \neg l_1) \\ \Box^*(l'_3 \Rightarrow \neg \boxed{a}\, \neg l_2) \\ \hline \Box^*(\mathbf{true} \Rightarrow \neg l'_1 \vee \neg l'_2 \vee \neg l'_3) \end{array}$$

[GEN1]
$$\begin{array}{c} \Box^*(l'_1 \Rightarrow \boxed{a}\, \neg l_1) \\ \vdots \\ \Box^*(l'_m \Rightarrow \boxed{a}\, \neg l_m) \\ \Box^*(l' \Rightarrow \neg \boxed{a}\, \neg l) \\ \Box^*(\mathbf{true} \Rightarrow l_1 \vee \ldots \vee l_m \vee \neg l) \\ \hline \Box^*(\mathbf{true} \Rightarrow \neg l'_1 \vee \ldots \vee \neg l'_m \vee \neg l') \end{array}$$

[GEN3]
$$\begin{array}{c} \Box^*(l'_1 \Rightarrow \boxed{a}\, \neg l_1) \\ \vdots \\ \Box^*(l'_m \Rightarrow \boxed{a}\, \neg l_m) \\ \Box^*(l' \Rightarrow \neg \boxed{a}\, \neg l) \\ \Box^*(\mathbf{true} \Rightarrow l_1 \vee \ldots \vee l_m) \\ \hline \Box^*(\mathbf{true} \Rightarrow \neg l'_1 \vee \ldots \vee \neg l'_m \vee \neg l') \end{array}$$

MRES is equivalent to classical resolution, as a formula and its negation cannot be true at the same state. The **GEN1** rule corresponds to generalisation (applied to $(\neg l_1 \wedge \ldots \wedge \neg l_m \Rightarrow \neg l)$, which is equivalent to the literal clause in the premises) and several applications of classical resolution. **GEN2** is a special case of **GEN1**, as the parent clauses can be resolved with tautologies as **true** $\Rightarrow l_1 \vee \neg l_1 \vee \neg l_2$. **GEN3** is similar to **GEN1**, however the negative modal clause is not resolved with the literal clause in the premises. Instead, the negative modal clause requires that resolution takes place between literals on the right-hand side of positive modal clauses and the literal clause. The resolvents in the inference rules **GEN1–GEN3** impose that the literals on the left-hand side of the modal clauses in the premises are not all satisfied whenever their conjunction leads to a contradiction in a successor state. Given the syntactic forms of clauses, the three rules are needed for completeness, as shown in [**?**].

Simplification. We assume standard simplification from classical logic to keep the clauses as simple as possible. For example, $D \vee l \vee l$ on the right-hand side of an initial or literal clause would be rewritten as $D \vee l$.

Example 2. We wish to establish whether the formula $\boxed{1}\boxed{2}(a \Rightarrow b) \Rightarrow (\boxed{1}\boxed{2}a \Rightarrow \boxed{1}\boxed{2}b)$ is valid in $\mathsf{K}_{(2)}$. The translation of its negation into the normal form is given by clauses (1)–(8) below. Then the inference rules are applied until **false** is generated. The full refutation follows.

1. **start** $\Rightarrow t_1$
2. $t_1 \Rightarrow \boxed{1}\, t_2$
3. $t_2 \Rightarrow \boxed{2}\, t_3$
4. **true** $\Rightarrow \neg t_3 \vee \neg a \vee b$

5. $t_1 \Rightarrow \boxed{1} t_4$

6. $t_4 \Rightarrow \boxed{2} a$

7. $t_1 \Rightarrow \neg \boxed{1} \neg t_5$

8. $t_5 \Rightarrow \neg \boxed{2} b$

9. $\mathbf{true} \Rightarrow \neg t_5 \vee \neg t_4 \vee \neg t_2$ [8,4,6,3 ,GEN1]

10. $\mathbf{true} \Rightarrow \neg t_1$ [9,7,5,2 ,GEN1]

11. **start** $\Rightarrow$ **false** [10,1,IRES1]

Clause (9) is obtained by an application of **GEN1** to clauses in the context of agent 2. Clause (10) is obtained by an application of the same rule, but in the context of agent 1. Clause (11) shows that a contradiction was found at the initial state. Therefore, the original formula is valid.

4 Clausal Resolution for Logics of Confluence

The inference rules of $\mathsf{K}_{(n)}$, given in Section 3.2, are resolution-based: whenever a set of (sub)formulae is identified as contradictory, the resolvents require that they are not all satisfied together. The extra inference rules for $\mathsf{K}_{(n)}^{p,q,r,s}$, with $p, q, r, s \in \{0, 1\}$, which we are about to present, have a different flavour: whenever we can identify that the set of clauses imply that $\Diamond_a^p \boxed{a}^q \psi$ holds, we add some new clauses that ensure that $\boxed{a}^r \Diamond_a^s \psi$ also holds. Because of the particular normal form we use here, there are, in fact, two general forms for the inference rules for $\mathsf{K}_{(n)}^{p,q,r,s}$, given in Table 2 (where l, l' are literals and C is a conjunction of literals).

$$[\mathbf{RES}_a^{p,1,r,s}]\ \frac{\Box^*(l \Rightarrow \boxed{a} l')}{\Box^*(\Diamond_a^p l \Rightarrow \boxed{a}^r \Diamond_a^s l')} \qquad [\mathbf{RES}_a^{p,0,r,s}]\ \frac{\Box^*(C \Rightarrow \Diamond_a^p l')}{\Box^*(C \Rightarrow \boxed{a}^r \Diamond_a^s l')}$$

Table 2. Inference Rules for $\mathbf{G}_a^{p,q,r,s}$

The conclusions of the inference rules in Table 2 are obtained from the semantics of the universal operator and the distribution axiom. For $\mathbf{RES}_a^{p,1,r,s}$, we have that $\Box^*(l \Rightarrow \boxed{a} l')$ is semantically equivalent to $\Box^*(\neg \boxed{a} l' \Rightarrow \neg l)$. By the definition of the universal operator, we obtain $\Box^*(\boxed{a}^p(\neg \boxed{a} l' \Rightarrow \neg l))$. Applying the distribution axiom $\mathbf{K}_a$ to this clause results in $\Box^*(\boxed{a}^p \neg \boxed{a} l' \Rightarrow \boxed{a}^p \neg l))$, which is semantically equivalent to $\Box^*(\neg \boxed{a}^p \neg l \Rightarrow \neg \boxed{a}^p \neg \boxed{a} l')$.

As $\langle a \rangle$ is an abbreviation for $\neg\boxed{a}\neg$ and because $\langle a \rangle^p\boxed{a}l'$ implies $\langle a \rangle^r\boxed{a}^s l'$ in $\mathsf{K}_{(n)}^{p,1,r,s}$, by classical reasoning, we have that $\square^*(\neg\boxed{a}^p\neg l \Rightarrow \neg\boxed{a}^p\neg\boxed{a}l')$ implies $\square^*(\langle a \rangle^p l \Rightarrow \langle a \rangle^r\boxed{a}^s l')$, the conclusion of $\mathbf{RES}_a^{p,1,r,s}$. The conclusion in the inference rule $\mathbf{RES}_a^{p,0,r,s}$ can be obtained in a similar way.

In order to respect the normal form, we also need to add clauses corresponding to the normal form of $\langle a \rangle^p l$ and $\langle a \rangle^s l'$, which occur in the conclusions of the inference rules. Let φ be a formula and let $\tau_0(\varphi)$ be the set of clauses resulting from the translation of φ into the normal form. Let $\mathcal{L}(\tau_0(\varphi))$ be the set of literals that might occur in the clause set, that is, $\mathcal{L}(\tau_0(\varphi)) = \{p, \neg p \mid$ for all p occurring in $\tau_0(\varphi)\}$. The set of *definition clauses* is given by

$$pos_{a,l} \Rightarrow \neg\boxed{a}\neg l$$
$$\neg pos_{a,l} \Rightarrow \boxed{a}\neg l$$

for all $l \in \mathcal{L}(\tau_0(\varphi))$, where $pos_{a,l}$ is a new propositional *definition symbol* used for renaming the negative modal literal $\langle a \rangle l$, that is, the definition clauses correspond to the normal form of $pos_{a,l} \Leftrightarrow \neg\boxed{a}\neg l$. We assume the set of definition clauses to be available whenever those symbols are used. We emphasise that those new propositional symbols and respective definition clauses may all be introduced at the beginning of the application of the resolution method. As no new propositional symbols are introduced by the inference rules, there is a finite number of clauses that might be expressed (modulo simplification) and, therefore, the clausal resolution method for each modal logic of confluence is terminating, thus convergent.

Table 3 shows the inference rules for each specific instance of $\mathbf{G}_a^{p,q,r,s}$, where $l, l' \in \mathcal{L}$; and D is a disjunction of literals. As $\mathbf{G}_a^{p,q,r,s}$ is semantically equivalent to $\mathbf{G}_a^{r,s,p,q}$, the inference rules for both systems are grouped together. Some of the rules are obtained straight from Table 2. For instance, the rules for reflexive systems, i.e. where $\mathbf{G}_a^{0,0,0,1}$ and $\mathbf{G}_a^{0,1,0,0}$ are valid. For other systems, the form of the inference rules might be slightly different from what would be obtained from a direct application of the general inference rules in Table 2. This is the case, for instance, for the inference rules for symmetric systems, that is, where $\mathbf{G}_a^{1,1,0,0}$ is valid. From Table 2, if a premise of the form $\square^*(l \Rightarrow \boxed{a}l')$, the conclusion is given by $\square^*(\langle a \rangle^{p=1} l \Rightarrow \boxed{a}^{r=0}\langle a \rangle^{s=0})$, which translates directly into the normal form as $\square^*(\mathbf{true} \Rightarrow pos_{a,l} \vee l')$. We have chosen, however, to translate the conclusion as $\square^*(\neg l' \Rightarrow \boxed{a}\neg l)$, which is semantically equivalent to the conclusion obtained by the general inference rule, but it does not use the new propositional symbols.

The inference rules given in Table 2 provide a systematic way of designing the inference rules for each specific modal logic of confluence. We note, however, that we might not need both inference rules in order to achieve a complete proof method for a particular logic. In the completeness proofs provided in the Appendix A, we show that the inference rules which introduce modalities in their conclusions from literal clauses (that is, the inference rules for $\mathbf{RES}_a^{0,0,r,s}$) are not needed for completeness. We also show that we only need one inference rule for logics in which $\mathbf{5}^{-1}$ is valid. In this case, we show that $\mathbf{RES}_a^{1,1,0,1}$ together with the resolution rules for $\mathsf{K}_{(n)}$ are sufficient for completeness. However we need both inference rules in the case of the proof method for logics in which **5** is valid.

Name	Resolution	
T	$[\mathbf{RES}_a^{0,0,0,1}]$	$\dfrac{\Box^*(\mathbf{true} \Rightarrow D \vee l)}{\Box^*(\neg D \Rightarrow \neg \boxed{a} \neg l)}$
	$[\mathbf{RES}_a^{0,1,0,0}]$	$\dfrac{\Box^*(l \Rightarrow \boxed{a} l')}{\Box^*(\mathbf{true} \Rightarrow \neg l \vee l')}$
Ban	$[\mathbf{RES}_a^{0,0,1,0}]$	$\dfrac{\Box^*(\mathbf{true} \Rightarrow D \vee l)}{\Box^*(\neg D \Rightarrow \boxed{a} l)}$
	$[\mathbf{RES}_a^{1,0,0,0}]$	$\dfrac{\Box^*(l \Rightarrow \neg \boxed{a} \neg l')}{\Box^*(\mathbf{true} \Rightarrow \neg l \vee l')}$
B	$[\mathbf{RES}_a^{0,0,1,1}]$	$\dfrac{\Box^*(\mathbf{true} \Rightarrow D \vee l)}{\Box^*(\neg D \Rightarrow \boxed{a} pos_{a,l})}$
	$[\mathbf{RES}_a^{1,1,0,0}]$	$\dfrac{\Box^*(l \Rightarrow \boxed{a} l')}{\Box^*(\neg l' \Rightarrow \boxed{a} \neg l)}$
D	$[\mathbf{RES}_a^{0,1,0,1}]$	$\dfrac{\Box^*(l \Rightarrow \boxed{a} l')}{\Box^*(l \Rightarrow \neg \boxed{a} \neg l')}$

Name	Resolution	
$\mathbf{5}^{-1}$	$[\mathbf{RES}_a^{0,1,1,1}]$	$\dfrac{\Box^*(l \Rightarrow \boxed{a} l')}{\Box^*(l \Rightarrow \boxed{a} pos_{a,l'})}$
	$[\mathbf{RES}_a^{1,1,0,1}]$	$\dfrac{\Box^*(l \Rightarrow \boxed{a} l')}{\Box^*(pos_{a,l} \Rightarrow \neg \boxed{a} \neg l')}$
F	$[\mathbf{RES}_a^{1,0,1,0}]$	$\dfrac{\Box^*(l \Rightarrow \neg \boxed{a} \neg l')}{\Box^*(l \Rightarrow \boxed{a} l')}$
5	$[\mathbf{RES}_a^{1,0,1,1}]$	$\dfrac{\Box^*(l \Rightarrow \neg \boxed{a} \neg l')}{\Box^*(l \Rightarrow \boxed{a} pos_{a,l'})}$
	$[\mathbf{RES}_a^{1,1,1,0}]$	$\dfrac{\Box^*(l \Rightarrow \boxed{a} l')}{\Box^*(pos_{a,l} \Rightarrow \boxed{a} l')}$
G1	$[\mathbf{RES}_a^{1,1,1,1}]$	$\dfrac{\Box^*(l \Rightarrow \boxed{a} l')}{\Box^*(pos_{a,l} \Rightarrow \boxed{a} pos_{a,l'})}$

Table 3. Resolution Rules

Given a formula φ in $\mathsf{K}_{(n)}^{p,q,r,s}$, with $p, q, r, s \in \{0, 1\}$, the resolution method for $\mathsf{K}_{(n)}$, given in Section 3, is applied to $\tau_0(\varphi)$ and the set of definition clauses. The extra inference rules for $\mathsf{K}_{(n)}^{p,q,r,s}$ cannot be applied to clauses if it generates nested new propositional symbols. For instance, $\mathbf{RES}_a^{1,1,1,1}$ cannot be applied to a clause as $\Box^*(l \Rightarrow \boxed{a} pos_{a,l'})$, as this would result in $\Box^*(pos_{a,l} \Rightarrow \boxed{a} pos_{a,pos_{a,l'}})$. This ensures that no new definition symbols are introduced by the method. The resolution-based proof method for each modal logic $\mathsf{K}_{(n)}^{p,q,r,s}$ is sound, complete, and terminating. The proofs are given in Appendix A. We present next just an example to illustrate the use of the method.

We note that a contradiction could also be derived if we had used the inference rule for **B**. The example shows, however, how one can combine the inference rules for systems for which both **T** and **5** are valid in order to achieve the same result.

Example 3. We show that $\varphi \stackrel{\text{def}}{=} p \Rightarrow \boxed{1} \Diamond_1 p$ is a valid formula in systems characterized by frames that are both reflexive and euclidean. Clauses (1)–(4) correspond to the translation of the negation of φ into the normal form.

$$
\begin{array}{rrll}
1. & \textbf{start} \Rightarrow t_0 & \\
2. & \textbf{true} \Rightarrow \neg t_0 \vee p & \\
3. & t_0 \Rightarrow \neg \boxed{1} \neg t_1 & \\
4. & t_1 \Rightarrow \boxed{1} \neg p & \\
5. & pos_{1,t_1} \Rightarrow \boxed{1} \neg p & [\mathbf{RES}_1^{1,1,1,0}, 4] \\
6. & \textbf{true} \Rightarrow \neg pos_{1,t_1} \vee \neg p & [\mathbf{RES}_1^{0,1,0,0}, 5] \\
7. & \neg pos_{1,t_1} \Rightarrow \boxed{1} \neg t_1 & [\mathbf{Def}.\ pos_{1,t_1}] \\
8. & \textbf{true} \Rightarrow \neg t_0 \vee pos_{1,t_1} & [\mathbf{MRES}, 7, 3] \\
9. & \textbf{true} \Rightarrow \neg t_0 \vee \neg p & [\mathbf{LRES}, 8, 6] \\
10. & \textbf{true} \Rightarrow \neg t_0 & [\mathbf{LRES}, 9, 2] \\
11. & \textbf{start} \Rightarrow \textbf{false} & [\mathbf{IRES1}, 10, 1]
\end{array}
$$

Clause (5) results from applying the euclidean inference rule to clause (4). Clause (6) results from applying the reflexive inference rule to (5). The definition clause for pos_{1,t_1} is introduced in (7). The remaining clauses are derived by the resolution calculus for $\mathsf{K}_{(1)}$. As a contradiction is found, given by clause (11), the set of clauses is unsatisfiable and the original formula φ is valid.

5 Closing Remarks

We have presented a sound, complete, and terminating proof method for logics of confluence, that is, normal multi-modal systems where axioms of the form

$$\mathbf{G}_a^{p,q,r,s} = \Diamond_a^p \boxed{a}^q \varphi \Rightarrow \boxed{a}^r \Diamond_a^s \varphi$$

are valid. The axioms $\mathbf{G}_a^{p,q,r,s}$ provide a general form for axioms widely used in logical formalisms applied to representation and reasoning within Computer Science. In this paper, we have restricted attention to the case where $p, q, r, s \in$

$\{0, 1\}$, but we believe that the proof method can be extended in a uniform way for dealing with the unsatisfiability problem for any values of p, q, r, and s, as far as we can identify the restrictions on the number of new propositional symbols introduced by the metod.

The proof method presented in this paper can be used to prove the unsatisfiability for eight families of logics and their combinations. However, as these are not dedicated proof methods, they might not provide the most efficient way of dealing with such a problem when considering a particular logic, as the method relies in the inference machine for $\mathsf{K}_{(n)}$. Note that the satisfiability problem for $\mathsf{K}_{(1)}$ is $PSPACE$-complete [**?**]. However, having a uniform approach for dealing with different logics means that implementation can be obtained in a straightforward way. Based on an implementation for $\mathsf{K}_{(n)}$, the automated reasoning for the remaining logics depends only on the implementation of few inference rules.

As mentioned, we intend to extend the method to deal with unrestricted values of p, q, r, and s. We also intend to investigate if the method can be applied together with other proof methods, as, for instance, tableaux.

A Correctness

In this appendix, we provide the correctness results related to the resolution-based calculus for modal logics of confluence, that is, soundness, termination, and completeness results for this method.

The proof that transformation of a formula $\varphi \in \mathsf{WFF}_{\mathsf{K}_{(n)}}$ into its normal form is satisfiability preserving is given in [**?**,**?**].

Soundness consists in showing that the application of the inference rules are satisfiability preserving, which follows from Lemmas 1 and 2 given below.

Lemma 1. $\mathbf{RES}_a^{p,1,r,s}$ *is sound.*

Proof. Let $\mathcal{M} = (\mathcal{W}, w_0, \pi, \mathcal{R}_1, \ldots, \mathcal{R}_n)$ be a model such that $\mathcal{M} \models \square^*(l \Rightarrow \boxed{a}\, l')$. By the semantics of the implication, we have that $\mathcal{M} \models \square^*(\neg\boxed{a}\, l' \Rightarrow \neg l)$. By the semantics of the universal operator, we obtain that $\mathcal{M} \models \square^*\boxed{a}^{\,p}(\neg\boxed{a}\, l' \Rightarrow \neg l)$. By axiom **K**, we have that $\mathcal{M} \models \square^*(\boxed{a}^{\,p}\neg\boxed{a}\, l' \Rightarrow \boxed{a}^{\,p}\neg l)$. By the semantics of implication, we obtain that $\mathcal{M} \models \square^*(\neg\boxed{a}^{\,p}\neg l \Rightarrow \neg\boxed{a}^{\,p}\neg\boxed{a}\, l')$. By $\mathbf{G}_a^{p,1,r,s}$ and classical reasoning, $\mathcal{M} \models \square^*(\neg\boxed{a}^{\,p}\neg l \Rightarrow \boxed{a}^{\,r}\neg\boxed{a}^{\,s}\neg l')$. By definition of $\Diamond_a$, $\mathcal{M} \models \square^*(\Diamond_a^{\,p} l \Rightarrow \boxed{a}^{\,r}\Diamond_a^{\,s} l')$. Therefore, if $\square^*(l \Rightarrow \boxed{a}\, l')$ is satisfiable, so it is $\square^*(\Diamond_a^{\,p} l \Rightarrow \boxed{a}^{\,r}\Diamond_a^{\,s} l')$. Thus, $\mathbf{RES}_a^{p,1,r,s}$ is sound.

Lemma 2. $\mathbf{RES}_a^{p,0,r,s}$ *is sound.*

Proof. Let $\mathcal{M} = (\mathcal{W}, w_0, \pi, \mathcal{R}_1, \ldots, \mathcal{R}_n)$ be a model such that $\mathcal{M} \models \square^*(C \Rightarrow \Diamond_a^{\,p} l')$. By the semantics of the universal operator, for all $w \in \mathcal{W}$, $\langle \mathcal{M}, w\rangle \models C \Rightarrow \Diamond_a^{\,p} l'$. By $\mathbf{G}_a^{p,0,r,s}$, we have that, for all $w \in \mathcal{W}$, $\langle \mathcal{M}, s\rangle \models \Diamond_a^{\,p} l' \Rightarrow \boxed{a}^{\,r}\Diamond_a^{\,s} l'$. By classical reasoning, we obtain $\langle \mathcal{M}, w\rangle \models (C \Rightarrow \boxed{a}^{\,r}\Diamond_a^{\,s} l')$, for all $w \in \mathcal{W}$. By the semantics of the universal operator, $\langle \mathcal{M}, w\rangle \models \square^*(C \Rightarrow \boxed{a}^{\,r}\Diamond_a^{\,s} l')$, for all $w \in \mathcal{W}$. So, $\langle \mathcal{M}, w_0\rangle \models \square^*(C \Rightarrow \boxed{a}^{\,r}\Diamond_a^{\,s} l')$ and, by definition of satisfiability, $\mathcal{M} \models \square^*(C \Rightarrow \boxed{a}^{\,r}\Diamond_a^{\,s} l')$. Therefore, if $\square^*(C \Rightarrow \neg\boxed{a}\neg l')$ is satisfiable, so it is $\square^*(C \Rightarrow \boxed{a}^{\,r}\Diamond_a^{\,s} l')$. Thus, $\mathbf{RES}_a^{p,0,r,s}$ is sound.

Termination is ensured by the fact that a given set of clauses contains only finitely many propositional symbols, from which only finitely many SNF_K clauses can be constructed and therefore only finitely many new SNF_K clauses can be derived. Note that, as the resolution rules $\mathbf{RES}_a^{p,q,r,s}$ cannot be applied to clauses which would result in nested definition symbols, all the corresponding definition clauses can be introduced at the beginning of the proof.

Completeness is proved by showing that if a given set of clauses is unsatisfiable, there is a refutation produced by the method presented here. The proof is by induction on the number of nodes of a graph, known as *behaviour graph*, built from a set of clauses. We prove that an empty behaviour graph corresponds to an unsatisfiable set of clauses and that, in this case, there is a refutation using the inference rules given in Section 3 and Table 2. The graph construction is similar to the construction of a canonical model, followed by filtrations based on the set of formulae (or clauses), often used to prove completeness for proof methods in modal logics (see [**?**], for instance, for definitions and examples). Intuitively, nodes in the graph correspond to states. Recall that for logics of confluence, the resolution calculus introduces a set of literals, which are used in the inference rules as new names for modal literals in the scope of the operator $\neg\boxed{a}\neg$, $a \in \mathcal{A}$. Therefore, nodes are defined as maximally consistent sets of literals and modal literals occuring in the set of clauses, including those literals introduced by definition clauses. That is, for any literal l occurring in the set of clauses and agents $a \in \mathcal{A}$, a node contains either l or $\neg l$; and either $\boxed{a}l$ or $\neg\boxed{a}l$. The set of edges correspond to the agents accessibility relations.

Formally, the graph for n agents is a tuple $\mathcal{G} = \langle \mathcal{N}, \mathcal{E}_1, \ldots, \mathcal{E}_n \rangle$, built from the set of SNF_K clauses $\mathcal{T}$, where $\mathcal{N}$ is a set of nodes and each $\mathcal{E}_a$ is a set of edges labelled by $a \in \mathcal{A}$. Intuitively, $\mathcal{N}$ corresponds to states, i.e., a consistent set of literals and modal literals occurring in $\mathcal{T}$. There are n types of edges representing the accessibility relations of each agent in $\mathcal{A}$. An edge labelled by $a \in \mathcal{A}$ is called an a-edge. Let η and η' be nodes. We say that η' is *a-reachable from* η, if there is a sequence of nodes $\eta_1, \eta_2, \ldots, \eta_k$ such that $\eta = \eta_1$ and $\eta' = \eta_k$ and $(\eta_j, \eta_{j+1}) \in \mathcal{E}_a$ for $j = 1, \ldots, k-1$. We say that η' is *immediately a-reachable* from η, if $(\eta, \eta') \in \mathcal{E}_a$. We say that the k-tuple $(\eta, \ldots, \eta') \in \mathcal{E}_a^k$, $k \in \mathbb{N}$, if there is a sequence of nodes $\eta_1, \ldots, \eta_k$, $\eta = \eta_1$ and $\eta' = \eta_k$, and for each η_j, $1 \leq j \leq k-1$, we have that $(\eta_j, \eta_{j+1}) \in \mathcal{E}_a$. Note that, if $k = 0$, then $\eta \in \mathcal{E}_a^0$, for all $\eta \in \mathcal{N}$ and $a \in \mathcal{A}$.

First, we define truth of a formula with respect to a set of literals and modal literals:

Definition 4. *Let $\mathcal{V}$ be a consistent set of literals and modal literals. Let φ, ψ, and ψ' be a Boolean combinations of literals and modal literals. We say that $\mathcal{V}$ satisfies φ (written $\mathcal{V} \models \varphi$), if, and only if:*

- *$\varphi \in \mathcal{V}$, if φ is a literal or a modal literal;*
- *φ is of the form $\psi \wedge \psi'$ and $\mathcal{V} \models \psi$ and $\mathcal{V} \models \psi'$;*
- *φ is of the form $\psi \vee \psi'$ and $\mathcal{V} \models \psi$ or $\mathcal{V} \models \psi'$;*
- *φ is of the form $\neg\psi$ and $\mathcal{V}$ does not satisfy ψ (written $\mathcal{V} \not\models \psi$).*

We define satisfiability of a formula and a set of formulae with respect to a node:

Definition 5. *Let $\mathcal{V}$ be a maximal consistent set of literals and modal literals, η be a node such that η contains all literals and modal literals in $\mathcal{V}$, φ be a Boolean combination of literals, and $\chi = \{\varphi_1, \ldots, \varphi_m\}$ be a set of formulae, where each φ_i, $1 \leq i \leq m$, is a Boolean combination of literals. We say that η* satisfies *φ (written $\eta \models \varphi$) if, and only if, $\mathcal{V} \models \varphi$. We say that η* satisfies *χ (written $\eta \models \chi$) if, and only if, $\eta \models \varphi_1 \wedge \ldots \wedge \varphi_m$.*

Let $\mathcal{T}$ be a set of clauses into SNF_K. We construct a finite direct *graph* $\mathcal{G} = \langle \mathcal{N}, \mathcal{E}_1, \ldots, \mathcal{E}_n \rangle$ *for* $\mathcal{T}$, where $\mathcal{N}$ is a set of nodes and each $\mathcal{E}_a$ is a set of a-edges, as follows. A node $\eta \in \mathcal{N}$ is a maximal consistent set of literals and modal literals, that is, it satisfies either a proposition or its negation; and it satisfies either a modal literal or its negation. Firstly, we delete any nodes that do not satisfy the literal clauses in $\mathcal{T}$, that is, if $\Box^*(\mathbf{true} \Rightarrow l_1 \vee \ldots \vee l_m) \in \mathcal{T}$, we delete the nodes $\eta \in \mathcal{N}$ such that $\eta \not\models l_1 \vee \ldots \vee l_m$. This ensures that literal clauses are satisfied by any node in $\mathcal{G}$. We also delete any nodes η such that $\eta \models pos_{a,l}$, but $\eta \not\models \Diamond_a l$. That is, we have that $pos_{a,l} \in \mathcal{V}$ if and only if $\Diamond_a l \in \mathcal{V}$ This ensures that all definition clauses are satisfied by all nodes in $\mathcal{G}$.

Let the initial states of the graph be those which satisfy all the right-hand sides of initial clauses. If all initial states are deleted, then the graph is empty.

Given a non-empty set of nodes, we construct the set of a-edges, $\mathcal{E}_a$, as follows. First, for each node η and for each agent $a \in \mathcal{A}$, let $\mathcal{C}_a^\eta \subseteq \mathcal{T}$ be the set of positive a-clauses corresponding to agent a, that is, the clauses of the form $\Box^*(l \Rightarrow \boxed{a}\, l')$, where l and l' are literals, whose left-hand side are satisfied by η. Let $\mathcal{L}_a^\eta$ be the set of literals in the scope of $\boxed{a}$ on the right-hand side from the clauses in $\mathcal{C}_a^\eta$, that is, if $\Box^*(l \Rightarrow \boxed{a}\, l') \in \mathcal{C}_a^\eta$, then $l' \in \mathcal{L}_a^\eta$. Build an a-edge from η to all the nodes η' that satisfy $\mathcal{L}_a^\eta$. Observe that when $\mathcal{C}_a^\eta$ is empty, then $\mathcal{L}_a^\eta$ is also empty. As an empty set of literals is satisfied in any node, there is an a-edge from η to every node in the graph, in order to ensure that the tautology $\mathbf{true} \Rightarrow \boxed{a}\,\mathbf{true}$ is satisfied. Note that this construction ensures that all positive a-clauses in $\mathcal{T}$ are satisfied. Next, consider any nodes that do not satisfy the negative a-clauses in $\mathcal{T}$. For each node η and for each agent $a \in \mathcal{A}$, if $\Box^*(l \Rightarrow \neg\boxed{a}\, l')$ is in $\mathcal{T}$, $\eta \models l$ and there is no a-edge between η and a node that satisfies $\neg l'$, then η is deleted.

The graph obtained after performing all possible deletions is called *reduced behaviour graph*.

We first show that a set of clauses is satisfiable if, and only if, the reduced graph for this set of clauses is non-empty.

Theorem 1. *Let $\mathcal{T}$ be a set of clauses. $\mathcal{T}$ is satisfiable in $\mathsf{K}_{(n)}$ if and only if the reduced behaviour graph $\mathcal{G}$ constructed from $\mathcal{T}$ is non-empty.*

Proof. ($\Rightarrow$) Assume that $\mathcal{T}$ is a satisfiable set of clauses. If we construct a graph from $\mathcal{T}$, we generate a node for each each maximal consistent set of literals and modal literals. Nodes are deleted only if they do not satisfy the set of literal clauses or definition clauses. Then we construct a-edges from each node to every other node, only deleting edges if the right-hand side of some positive a-clause is not satisfied. Similarly nodes are deleted if negative a-clauses cannot be satisfied. Hence a satisfiable set of clauses will result in a non-empty graph.

($\Leftarrow$) Assume that the reduced graph $\mathcal{G} = \langle \mathcal{N}, \mathcal{E}_1, \ldots, \mathcal{E}_n \rangle$ constructed from $\mathcal{T}$ is non-empty. To show that $\mathcal{T}$ is satisfiable we construct a model $\mathcal{M}$ from $\mathcal{G}$. Let $\mathcal{M} = \langle \mathcal{W}, \pi, \mathcal{R}_1, \ldots, \mathcal{R}_n \rangle$. Given the set of propositions occuring in the set of clauses $\mathcal{T}$, $\mathcal{P}_{\mathcal{T}}$, let $s_i \in \mathcal{N}$, where $0 \leqslant i \leqslant 2^{|\mathcal{P}_{\mathcal{T}}|} - 1$, there is a function $node : \mathcal{N} \rightarrow \mathcal{W}$ mapping each consistent set of literals and modal literals to names of nodes such that $node(\eta') = s_0$ for η' some initial node and each node is mapped to a different name. Let $\mathcal{R}_a = \mathcal{E}_a$ and let $\pi(s_j)(p) = true$ if, and only if, $node(\eta) = s_j$ and $p \in \eta$.

Theorem 2. *Let $\mathcal{T}$ be an unsatisfiable set of clauses in $\mathsf{K}_{(n)}$. A contradiction can be derived by applying the resolution rules given in Section 3.*

Proof. Given a set of clauses $\mathcal{T}$, construct a reduced behaviour graph as described above.

First assume that the initial and literal clauses are unsatisfiable. Thus all initial nodes will be removed from the reduced graph and the graph becomes empty. From the completeness of classical resolution there is a series of resolution steps which can be applied to the right-hand side of these clauses which lead to the derivation of **false**. We can mimic these steps by applying the IRES1, IRES2 or LRES resolution rules to the initial and literal clauses to derive **start** $\Rightarrow$ **false** or **true** $\Rightarrow$ **false**.

Next, if the non-reduced graph is not empty, consider any nodes that do not satisfy the negative a-clauses in $\mathcal{T}$. For each node η and for each agent $a \in \mathcal{A}$, if $\Box^*(l \Rightarrow \neg \boxed{a} l')$ is in $\mathcal{T}$, $\eta \models l$ and there is no a-edge between η and a node that satisfies $\neg l'$, then η is deleted.

Let $\mathcal{C}_a^\eta$ in $\mathcal{T}$ be the set of positive a-clauses corresponding to agent a, that is, the clauses of the form $\Box^*(l_j \Rightarrow \boxed{a} l'_j)$, where l_j and l'_j are literals, whose left-hand side are satisfied by η. Let $\mathcal{L}_a^\eta$ be the set of literals in the scope of $\boxed{a}$ on the right-hand side from the clauses in $\mathcal{C}_a^\eta$, that is, if $\Box^*(l_j \Rightarrow \boxed{a} l'_j) \in \mathcal{C}_a^\eta$, then $l'_j \in \mathcal{L}_a^\eta$. From the construction of the graph, for a clause $\Box^*(l \Rightarrow \neg \boxed{a} l')$, if $\eta \models l$ but there is no a-edge to a node containing $\neg l'$, it means that $\neg l'$, $\mathcal{L}_a^\eta$, and the right-hand side of the literal clauses must be contradictory.

First assume that $\neg l'$ and $\mathcal{L}_a^\eta$ is contradictory. Then, $\mathcal{C}_a^\eta$ in $\mathcal{T}$ contains a clause as $\Box^*(l_1 \Rightarrow \boxed{a} l')$ where, from the definition of $\mathcal{C}_a$, $\eta \models l_1$. Thus, by an application of **MRES** to this clause and $l \Rightarrow \neg\boxed{a}l'$, we derive $\Box^*(\textbf{true} \Rightarrow \neg l_1 \vee \neg l)$ and η is removed as required.

Next assume that $\mathcal{L}_a^\eta$ itself is contradictory. This means there must be clauses of the form $\Box^*(l_1 \Rightarrow \boxed{a}l'')$, $\Box^*(l_2 \Rightarrow \boxed{a}\neg l'') \in \mathcal{C}_a^\eta$, where $\eta \models l_1$ and $\eta \models l_2$. Thus we can apply **GEN2** to these clauses and the negative modal clause $l \Rightarrow \neg\boxed{a}l'$ deriving $\Box^*(\textbf{true} \Rightarrow \neg l_1 \vee \neg l_2 \vee \neg l)$. Hence the addition of this resolvent means that η will be deleted as required.

Next assume that $\neg l'$ and the right-hand side of the literal clauses are contradictory. The case where the right-hand sides of the literal clauses themselves are contradictory has been covered above (by applying **LRES**). By applying **LRES** to the set of literal clauses, we obtain $\Box^*(\textbf{true} \Rightarrow l')$ and use this with $\Box^*(l \Rightarrow \neg\boxed{a}l')$ to apply **GEN1** and generate $\Box^*(\textbf{true} \Rightarrow \neg l)$ which will delete η as required.

If $\mathcal{L}_a^\eta$ and the right-hand side of literal clauses all contribute to the contradiction (but not $\neg l'$), applying **GEN3** to the relevant clauses will delete η as required.

Finally we assume that $\neg l'$ and $\mathcal{L}_a^\eta$ and the right-hand side of the literal clauses all contribute to the contradiction. Thus, similarly to the above, applying **GEN1** to the relevant clauses will delete η as required.

Summarising, **IRES1**, **IRES2** and **LRES** remove from the graph nodes related to contradictions in the set of literal clauses. The rule **MRES** also simulates classical resolution, removing from the graph those nodes related to contradiction within the set of modal literals. The inference rule **GEN1** deletes parts of the graph related to contradictions between the literal in the scope of $\neg\boxed{a}\neg$, the set of literal clauses, and the literals in the scope of $\boxed{agent}$The resolution rule **GEN2** deletes parts of the graph related to contradictions between the the literals in the scope of $\boxed{agent}$Finally, **GEN3** deletes parts of the graph related to contradictions between the the literals in the scope of $\boxed{agent}$and the set of literal clauses. These are all possible combinations of contradicting sets withing a clause set.

If the resulting graph is empty, the set of clauses $\mathcal{T}$ is not satisfiable and there is a resolution proof corresponding to the deletion procedure, as described above. If the graph is not empty, it provides a model for the satisfiable set of clauses $\mathcal{T}$.

After exhaustively appling deletions to the graph, if the graph is empty, by completeness of $\mathsf{RES_K}$, there is a proof by the resolution rules shown in Section 3. If the graph is not empty, we have to check whether we can build a model for $\mathcal{T}$, where $\mathbf{G}_a^{p,q,r,s}$ holds. The fact that this is possible is given by the following lemmas.

Lemma 3. *Let $\mathcal{T}$ be an unsatisfiable set of clauses in reflexive systems. A contradiction can be derived by applying the resolution rules given in Section 3 and* $\mathbf{RES}_a^{0,1,0,0}$.

Proof. Consider any normal modal logic where each binary relation $\mathcal{R}_a$ is reflexive. We construct a graph $\mathcal{G} = \langle \mathcal{N}, \mathcal{E}_1, \ldots, \mathcal{E}_n \rangle$ for $\mathsf{K}_{(n)}$ as described above. We show that by applying the resolution rule $\mathbf{RES}_a^{0,1,0,0}$ any non-reflexive node is deleted. Consider a node η and $a \in \mathcal{A}$ and $(\eta, \eta) \notin \mathcal{E}_a$. There are two options: there are no a-edges out of η; or a-edges lead from η but there is no a-edge from η to itself.

For the former, from the construction of the graph $\mathcal{G}$, we have tried to construct as many edges as possible. That is there must be some a-clauses of the form $\Box^*(l'_1 \Rightarrow \boxed{a} l_1)$, $\Box^*(l'_2 \Rightarrow \boxed{a} l_2)$, $\ldots$, $\Box^*(l'_k \Rightarrow \boxed{a} l_k)$, such that for each $j = 1, \ldots, k$, $\eta \models l'_j$ and either $\bigwedge_j l_j$ is contradictory (e.g. when $l_j = \neg l_h$ for $j, h = 1, \ldots, k$) or when $\bigwedge_j l_j$ and the set of clauses from the right-hand side of the literal clauses is contradictory. Note that we assume that this node does not have any unsatisfied negative a-clauses as such a node would have been previously deleted by the related deletion rule. For the case $\bigwedge_j l_j$ is contradictory there must be two clauses $\Box^*(l'_1 \Rightarrow \boxed{a} l_1)$ and $\Box^*(l'_2 \Rightarrow \boxed{a} \neg l_1)$ such that $\eta \models l'_1$ and $\eta \models l'_2$. Applying $\mathbf{RES}_a^{0,1,0,0}$ to each we obtain $\Box^*(\mathbf{true} \Rightarrow \neg l'_1 \vee l_1)$ and $\Box^*(\mathbf{true} \Rightarrow \neg l'_2 \vee \neg l_1)$ and by applying LRES we can obtain $\Box^*(\mathbf{true} \Rightarrow \neg l'_1 \vee \neg l'_2)$. Adding this to the graph deletes η. The case where $\bigwedge_j l_j$ and the set of clauses from the right-hand side of the literal clauses is contradictory is similar. By the completeness of classical resolution we can again derive a clause that removes η.

Next consider the second case for some node η. As we have attempted to construct as many edges as possible from every node there must be a clause $\Box^*(l'_1 \Rightarrow \boxed{a} l_1)$ such that $\eta \models l'_1$ and $\eta \not\models l_1$. By applying $\mathbf{RES}_a^{0,1,0,0}$ we obtain $\Box^*(\mathbf{true} \Rightarrow \neg l'_1 \vee l_1)$. As $\eta \not\models \neg l'_1 \vee l_1$, η is deleted as required.

Note that $\mathbf{RES}_a^{0,0,0,1}$ is not required for completeness as it can be simulated by other inference rules. Assume that $\Box^*(\mathbf{true} \Rightarrow D \vee l)$ is in $\mathcal{T}$. Recall that for all literals l occurring in $\mathcal{T}$, the definition clauses $\Box^*(pos_{a,l} \Rightarrow \neg \boxed{a} \neg l)$ and $\Box^*(\neg pos_{a,l} \Rightarrow \boxed{a} \neg l)$ are also in $\mathcal{T}$. Applying $\mathbf{RES}_a^{0,1,0,0}$ to $\Box^*(\neg pos_{a,l} \Rightarrow \boxed{a} \neg l)$ results in $\Box^*(\mathbf{true} \Rightarrow pos_{a,l} \vee \neg l)$. Applying **LRES** to $\Box^*(\mathbf{true} \Rightarrow pos_{a,l} \vee \neg l)$ and $\Box^*(\mathbf{true} \Rightarrow D \vee l)$ results in $\Box^*(\mathbf{true} \Rightarrow D \vee pos_{a,l})$, which is semantically equivalent to $\Box^*(\neg D \Rightarrow \neg \boxed{a} \neg l)$, the resolvent of $\mathbf{RES}_a^{0,0,0,1}$ from $\Box^*(\mathbf{true} \Rightarrow D \vee l)$.

Lemma 4. *Let $\mathcal{T}$ be an unsatisfiable set of clauses in modally banal systems. A contradiction can be derived by applying the resolution rules given in Section 3 and* $\mathbf{RES}_a^{1,0,0,0}$.

Proof. Consider any normal modal logic where each binary relation $\mathcal{R}_a$ is modally banal. We construct a graph $\mathcal{G} = \langle \mathcal{N}, \mathcal{E}_1, \ldots, \mathcal{E}_n \rangle$ for $\mathsf{K}_{(n)}$ as described above. We show that by applying the resolution rule $\mathbf{RES}_a^{1,0,0,0}$ any node that does not satisfy the frame conditions is deleted. Consider a node η and $a \in \mathcal{A}$, $(\eta, \eta') \in \mathcal{E}_a$, but $\eta \neq \eta'$.

First note that if $\eta \neq \eta'$, there must be a literal l such that $\eta \models l$ and $\eta' \not\models l$. As there is an edge from η to η', we have that $\eta \models pos_{a,\neg l}$. Applying $\mathbf{RES}_a^{1,0,0,0}$ to $\Box^*(pos_{a,\neg l} \Rightarrow \neg \boxed{a} l)$ we obtain $\Box^*(\mathbf{true} \Rightarrow \neg pos_{a,\neg l} \vee \neg l)$, which is not satisfied in η. Therefore, η is deleted as required.

Assume that $\Box^*(l \Rightarrow \neg \boxed{a} \neg l') \in \mathcal{T}$ and $\eta \models l$. As all modal negative modal clauses are satisfied in the graph, there must be an edge (η, η) in $\mathcal{E}_a$, otherwise the node would have been removed. If $\eta \not\models l'$, by adding $\Box^*(\mathbf{true} \Rightarrow \neg l \vee l')$, the resolvent of $\mathbf{RES}_a^{1,0,0,0}$ from $\Box^*(l \Rightarrow \neg \boxed{a} \neg l')$, we have that η is deleted. This deletion corresponds to applications of **LRES** to $\Box^*(\mathbf{true} \Rightarrow \neg l \vee l')$ and the set of literal clauses that together imply $\neg l'$.

Note that $\mathbf{RES}_a^{0,0,1,0}$ is not required for completeness as it can be simulated by other inference rules. Assume that $\Box^*(\mathbf{true} \Rightarrow D \vee l)$ is in $\mathcal{T}$. Recall that for all literals l occurring in $\mathcal{T}$, the definition clause $\Box^*(pos_{a,\neg l} \Rightarrow \neg \boxed{a} l)$ is also in $\mathcal{T}$. Applying $\mathbf{RES}_a^{1,0,0,0}$ to $\Box^*(pos_{a,\neg l} \Rightarrow \neg \boxed{a} l)$ results in $\Box^*(\mathbf{true} \Rightarrow \neg pos_{a,\neg l} \vee \neg l)$. Applying **LRES** to $\Box^*(\mathbf{true} \Rightarrow D \vee l)$ and $\Box^*(\mathbf{true} \Rightarrow \neg pos_{a,\neg l} \vee \neg l)$ results in $\Box^*(\mathbf{true} \Rightarrow D \vee \neg pos_{a,\neg l})$, which is semantically equivalent to $\Box^*(\neg D \Rightarrow \boxed{a} l)$, the resolvent of $\mathbf{RES}_a^{0,0,1,0}$ from $\Box^*(\mathbf{true} \Rightarrow D \vee l)$ is in $\mathcal{T}$.

Lemma 5. *Let $\mathcal{T}$ be an unsatisfiable set of clauses in symmetric systems. A contradiction can be derived by applying the resolution rules given in Section 3 and* $\mathbf{RES}_a^{1,1,0,0}$.

Proof. Consider any normal modal logic where each binary relation $\mathcal{R}_a$ is symmetric. We construct a graph $\mathcal{G} = \langle \mathcal{N}, \mathcal{E}_1, \ldots, \mathcal{E}_n \rangle$, as described for $\mathsf{K}_{(n)}$. We show that by applying the inference rule $\mathbf{RES}_a^{1,1,0,0}$ any non-symmetric edge between two nodes is deleted. Consider any pair of nodes η and η' such that there is some $a \in \mathcal{A}$ and $(\eta, \eta') \in \mathcal{E}_a$ but $(\eta', \eta) \notin \mathcal{E}_a$.

From the construction of the graph $\mathcal{G}$, we have tried to construct as many edges as possible. That is, there must be some positive a-clause of the form $\Box^*(l' \Rightarrow \boxed{a} l)$, such that $\eta' \models l'$ and $\eta \not\models l$ (i.e. $\eta \models \neg l$). Applying $\mathbf{RES}_a^{1,1,0,0}$, we obtain $\Box^*(\neg l \Rightarrow \boxed{a} \neg l')$. As $\eta \models \neg l$ and $\eta' \models l'$, from the construction of the graph, the resolvent of SYM removes the edge (η, η') from $\mathcal{E}_a$ as required.

Note that $\mathbf{RES}_a^{0,0,1,1}$ is not required for completeness of the proof method for symmetric systems. Assume that $\Box^*(\mathbf{true} \Rightarrow D \vee l)$ is in $\mathcal{T}$. By applying $\mathbf{RES}_a^{0,0,1,1}$ to $\Box^*(\mathbf{true} \Rightarrow D \vee l)$ we obtain $\Box^*(\neg D \Rightarrow \boxed{a} pos_{a,l})$, which removes from the graph all edges from nodes that satisfy $\neg D$ to nodes that do not satisfy $pos_{a,l}$. Assume $(\eta, \eta') \in \mathcal{E}_a$, where $\eta \models \neg D$, but $\eta' \not\models pos_{a,l}$. If $\eta' \not\models pos_{a,l}$, then there must be a clause $\Box^*(l_1 \Rightarrow \boxed{a}\neg l)$, such that $\eta' \models l_1$ and all edges from η' to nodes which satisfy l were removed during the construction of the graph. By applying $\mathbf{RES}_a^{1,1,0,0}$ to $\Box^*(l_1 \Rightarrow \boxed{a}\neg l)$ we obtain $\Box^*(\neg l \Rightarrow \boxed{a}\neg l_1)$. Applying **MRES** to this and to the definition clause $\Box^*(pos_{a,l_1} \Rightarrow \neg\boxed{a}\neg l_1)$, we obtain $\Box^*(\mathbf{true} \Rightarrow \neg l \vee \neg pos_{a,l_1})$. Applying **LRES** to $\Box^*(\mathbf{true} \Rightarrow D \vee l)$ and $\Box^*(\mathbf{true} \Rightarrow \neg l \vee \neg pos_{a,l_1})$, we obtain $\Box^*(\mathbf{true} \Rightarrow D \vee \neg pos_{a,l_1})$, which removes the edges from η, which satisfies $\neg D$, to η', which satisfies l_1. Therefore, $\mathbf{RES}_a^{0,0,1,1}$ is not needed for completeness.

Lemma 6. *Let $\mathcal{T}$ be an unsatisfiable set of clauses in serial systems. A contradiction can be derived by applying the resolution rules given in Section 3 and* $\mathbf{RES}_a^{0,1,0,1}$.

Proof. Consider any normal modal logic where each binary relation $\mathcal{R}_a$ is serial. We construct a graph $\mathcal{G} = \langle \mathcal{N}, \mathcal{E}_1, \ldots, \mathcal{E}_n \rangle$ for $\mathsf{K}_{(n)}$, as described before. We show that by applying the resolution rule $\mathbf{RES}_a^{0,1,0,1}$ any non-serial node is deleted. Consider any node η such that there is some $a \in \mathcal{A}$ and there is no η' such that $(\eta, \eta') \in \mathcal{E}_a$.

From the construction of the graph $\mathcal{G}$, we have tried to construct as many edges as possible. That is there must be positive a-clauses of the form $\Box^*(l'_1 \Rightarrow \boxed{a} l_1)$, $\Box^*(l'_2 \Rightarrow \boxed{a} l_2)$, ..., $\Box^*(l'_k \Rightarrow \boxed{a} l_k)$, such that for each $j = 1, \ldots, k$, $\eta \models l'_j$ but no node satisfies both $\bigwedge_j l_j$ and the set of literal clauses. By applying $\mathbf{RES}_a^{0,1,0,1}$, we add $\Box^*(l'_j \Rightarrow \neg\boxed{a}\neg l_j)$ for $j = 1, \ldots, k$. From the construction of the graph η does not satisfy these clauses and η is removed. Applying **MRES**, **GEN1**, **GEN2** or **GEN3** (as described in the completeness argument for $\mathsf{K}_{(n)}$) will achieve the deletion of η.

Lemma 7. *Let $\mathcal{T}$ be an unsatisfiable set of clauses in 5^{-1} systems. A contradiction can be derived by applying the resolution rules given in Section 3 and* $\mathbf{RES}_a^{0,1,1,1}$.

Proof. Consider any normal modal logic where each binary relation $\mathcal{R}_a$ respects the frame conditions for $\mathbf{G}_a^{1,1,1,0}$. We construct a graph $\mathcal{G} = \langle \mathcal{N}, \mathcal{E}_1, \ldots, \mathcal{E}_n \rangle$ for $\mathsf{K}_{(n)}$ as described above. We show that by applying the resolution rule $\mathbf{RES}_a^{0,1,1,1}$ any edge that does not satisfy the frame conditions is deleted.

Consider nodes η, η' in $\mathcal{E}_a$ for some $a \in \mathcal{A}$, where $(\eta, \eta') \in \mathcal{E}_a$, but there is no η'' such that (η, η'') and (η', η'') are in $\mathcal{E}_a$. If there is no such η'', then $\mathcal{L}_a^{\eta} \cup \mathcal{L}_a^{\eta'}$ is contradictory. Thus, there must be clauses $\Box^*(l_1 \Rightarrow \boxed{a} l)$

and $\square^*(l_2 \Rightarrow \boxed{a}\neg l)$ such that $\eta \models l_1$ and $\eta' \models l_2$. Recall that the definition clause $\square^*(pos_{a,l} \Rightarrow \neg\boxed{a}\neg l)$ is in the set of clauses. Applying **MRES** to $\square^*(pos_{a,l} \Rightarrow \neg\boxed{a}\neg l)$ and $\square^*(l_2 \Rightarrow \boxed{a}\neg l)$ results in $\square^*(\textbf{true} \Rightarrow l_2 \vee \neg pos_{a,l})$. Applying $\mathbf{RES}_a^{0,1,1,1}$ to $\square^*(l_1 \Rightarrow \boxed{a}l)$ results in $\square^*(l_1 \Rightarrow \boxed{a}pos_{a,l})$. As $\square^*(pos_{a,l_2} \Rightarrow \neg\boxed{a}l_2)$ is in the set of clauses, by applying **GEN1** to $\square^*(l_1 \Rightarrow \boxed{a}pos_{a,l})$, $\square^*(pos_{a,l_2} \Rightarrow \neg\boxed{a}l_2)$, and $\square^*(\textbf{true} \Rightarrow l_2 \vee \neg pos_{a,l})$ we obtain $\square^*(\textbf{true} \Rightarrow \neg l_1 \vee \neg pos_{a,l_2})$, which is equivalent to $\square^*(l_1 \Rightarrow \boxed{a}\neg l_2)$, which removes the edge (η, η') as required.

Note that $\mathbf{RES}_a^{1,1,0,1}$ is not required for completeness. Assume that $\square^*(l \Rightarrow \boxed{a}l')$ is in the set $\mathcal{T}$ of clauses. Applying $\mathbf{RES}_a^{1,1,0,1}$ to $\square^*(l \Rightarrow \boxed{a}l')$ results in $\square^*(pos_{a,l} \Rightarrow \neg\boxed{a}\neg l')$, which removes from the graph all nodes that satisfy $pos_{a,l}$ but do not satisfy $\neg\boxed{a}\neg l'$. Assume $(\eta, \eta') \in \mathcal{E}_a$, where $\eta' \models l$. By construction of the graph, as $(\eta, \eta') \in \mathcal{E}_a$, we have that $\eta \models pos_{a,l}$.

If there are no edges out of η', then, by construction of the graph, we have that $\eta' \models \neg pos_{a,l'}$. Now, applying $\mathbf{RES}_a^{0,1,1,1}$ to $\square^*(l \Rightarrow \boxed{a}l')$ results in $\square^*(l \Rightarrow \boxed{a}pos_{a,l'})$. As $\eta' \not\models pos_{a,l'}$, the edge (η, η') is removed from the graph. As $pos_{a,l'}$ is no longer satisfied at η, η is removed from the graph.

If there are edges $(\eta', \eta'') \in \mathcal{E}_a$, but there are no edge (η, η''), then there must be a clause as $\square^*(l_1 \Rightarrow \boxed{a}l_2)$ such that $\eta \models l_1$ and $\eta'' \not\models l_2$. Moreover, η' satisfies $\boxed{a}\neg l_2$ and $\neg pos_{a,l_2}$, otherwise the edge relation would meet the frame conditions for 5^{-1} systems. Applying $\mathbf{RES}_a^{0,1,1,1}$ to $\square^*(l_1 \Rightarrow \boxed{a}l_2)$ results in $\square^*(l_1 \Rightarrow \boxed{a}pos_{a,l_2})$. As $\eta' \not\models pos_{a,l_2}$, the edge (η, η') is removed from the graph. Because $pos_{a,l}$ is no longer satisfied at η, the node is removed from the graph.

We disallow the application of $\mathbf{RES}_a^{0,1,1,1}$ to clauses whose right-hand sides are of the form $\boxed{a}pos_{a,l'}$. Applying $\mathbf{RES}_a^{0,1,1,1}$ to a clause as $\square^*(l \Rightarrow \boxed{a}pos_{a,l'})$ is not necessary for completeness and might cause the method to be non terminating, as an unrestricted number of *nested* literals as $pos_{a,pos_{a,l'}}$ could be generated. Assume that $\square^*(l \Rightarrow \boxed{a}pos_{a,l'})$ is in the set of clauses and that there is a node η such that $\eta \models l$. If there are no edges out of η, then we have that both $pos_{a,pos_{a,l'}}$ and $pos_{a,l'}$ are not satisfied at η. If there is a node η', such that (η, η') in $\mathcal{E}_a$, then by construction of the graph we have that $\eta' \models pos_{a,l'}$, otherwise the edge (η, η') would have been removed from the graph. There are several cases:

1. If $\square^*(l \Rightarrow \boxed{a}pos_{a,l'})$ was obtained by an application of $\mathbf{RES}_a^{0,1,1,1}$ to the clause $\square^*(l \Rightarrow \boxed{a}l')$, we have that $\eta \models \boxed{a}pos_{a,l'}$. As $(\eta, \eta') \in \mathcal{E}_a$, we have that $\eta' \models pos_{a,l'}$ and, by construction of the graph, we also have that there must exist a node η'' such that both (η, η'') and (η', η'') are in $\mathcal{E}_a$, otherwise the frame conditions for 5^{-1} would not have been met and we would have

the edge (η, η') would have been removed from the graph. As $\Box^*(l \Rightarrow \boxed{a} pos_{a,l'})$ is in $\mathcal{T}$, $\eta \models l$, and $(\eta, \eta'') \in \mathcal{E}_a$, we have that $\eta'' \models pos_{a,l'}$. As $(\eta', \eta'') \in \mathcal{E}_a$, we have that $\eta'' \models \neg\boxed{a}\neg pos_{a,l'}$, that is, $pos_{a,pos_{a,l'}}$ holds exactly where $pos_{a,l'}$ holds.

2. If $\Box^*(l \Rightarrow \boxed{a} pos_{a,l'})$ was obtained by an application of $\mathbf{RES}_a^{1,0,1,1}$ to the clause $\Box^*(l \Rightarrow \neg\boxed{a}\neg l')$, then $pos_{a,l'}$ holds at η. Because (η, eta') there must exist a node η'' such that both (η, η'') and (η', η'') are in $\mathcal{E}_a$, otherwise the frame conditions for 5^{-1} would not have been met and we would have the edge (η, η') would have been removed from the graph. As $\Box^*(l \Rightarrow \boxed{a} pos_{a,l'}) \in \mathcal{T}$, $\eta \models l$, and $(\eta, \eta'') \in \mathcal{E}_a$, we have that $\eta'' \models pos_{a,l'}$. As $(\eta, \eta'') \in \mathcal{E}_a$, we have that $\eta \models \neg\boxed{a}\neg pos_{a,l'}$, that is, $pos_{a,pos_{a,l'}}$ holds exactly where $pos_{a,l'}$ holds.
3. If $\Box^*(pos_a \Rightarrow \boxed{a} pos_{a,l'})$ was obtained by an application of $\mathbf{RES}_a^{1,1,1,1}$ to the clause $\Box^*(l \Rightarrow \boxed{a} l')$: **to be continued.**
4. If $\Box^*(l \Rightarrow \boxed{a} pos_{a,l'})$ was obtained by an application of $\mathbf{RES}_a^{1,1,0,0}$ to the clause $\Box^*(\neg pos_{a,l} \Rightarrow \boxed{a}\neg l)$: **to be continued.**

Lemma 8. *Let $\mathcal{T}$ be an unsatisfiable in functional systems. A contradiction can be derived by applying the resolution rules given in Section 3 and $\mathbf{RES}_a^{1,0,1,0}$.*

Proof. Consider any normal modal logic where each binary relation $\mathcal{R}_a$ is functional. We construct a graph $\mathcal{G} = \langle \mathcal{N}, \mathcal{E}_1, \ldots, \mathcal{E}_n \rangle$ for $\mathsf{K}_{(n)}$ as described above. We show that by applying the resolution rule $\mathbf{RES}_a^{1,0,1,0}$ any edge that does not satisfy the frame conditions is deleted. Consider nodes η, η', and η'' in $\mathcal{E}_a$ for some $a \in \mathcal{A}$, where $(\eta, \eta'), (\eta, \eta'') \in \mathcal{E}_a$, but $\eta' \neq \eta''$.

If $\eta' \neq \eta''$, then there must be a literal, say l, such that $\eta' \models l$ and $\eta'' \not\models l$. Because $(\eta, \eta') \in \mathcal{E}_a$ and $\eta' \models l$, we have that $\eta \models pos_{a,l}$. By construction, all nodes satisfy the definition clauses. In particular, $\eta \models pos_{a,l} \Rightarrow \neg\boxed{a}\neg l$. Applying $\mathbf{RES}_a^{1,0,1,0}$ to $\Box^*(pos_{a,l} \Rightarrow \neg\boxed{a}\neg l)$ results in $\Box^*(pos_{a,l} \Rightarrow \boxed{a} l)$. By construction of the graph, as $\eta'' \models \neg l$, the edge (η, η'') is removed from the graph, as required.

Lemma 9. *Let $\mathcal{T}$ be an unsatisfiable set if clauses in Euclidean systems. A contradiction can be derived by applying the resolution rules given in Section 3, $\mathbf{RES}_a^{1,0,1,1}$, and $\mathbf{RES}_a^{1,1,1,0}$.*

Proof. Consider any normal modal logic where each binary relation $\mathcal{R}_a$ is Euclidean. We construct a graph $\mathcal{G} = \langle \mathcal{N}, \mathcal{E}_1, \ldots, \mathcal{E}_n \rangle$, as described in Subsection A for $\mathsf{K}_{(n)}$. We show that by applying the rules $\mathbf{RES}_a^{1,0,1,1}$ and $\mathbf{RES}_a^{1,1,1,0}$ any non-Euclidean edges are deleted.

Suppose that $\Box^*(l_1 \Rightarrow \neg\boxed{a}\neg l)$ is in the set of clauses. Then, there must be nodes η and $\eta' \in \mathcal{N}$, such that $(\eta, \eta') \in \mathcal{E}_a$, $\eta \models l_1$, and $\eta' \models l$ (otherwise the negative a-clause would not be satisfied and η would have been removed during the graph construction). Now, suppose that there is no a-edge from η' to a node η'' that satisfies l. Then, the a-edge relation is not Euclidean, because $\boxed{a}\neg\boxed{a}\neg l$ does not hold at η. Applying the rule $\mathbf{RES}_a^{1,0,1,1}$, we obtain the clause $\square^*(l_1 \Rightarrow \boxed{a} pos_{a,l})$. Because $\eta \models l_1$, we obtain, $\eta \models \boxed{a} pos_{a,l}$. Therefore, by construction of the graph, any edges from η to nodes that do not satisfy $pos_{a,l}$ are removed from the graph. In particular, as $\eta' \not\models pos_{a,l}$, the edge (η, η') is removed from the graph, as required.

Consider nodes $\eta, \eta', \eta'' \in \mathcal{N}$ such that (η, η') and (η, η'') are a-edges in $\mathcal{E}_a$, but (η', η'') is not an a-edge in $\mathcal{E}_a$. Thus the a-edge relation is not Euclidean. From the graph construction we have tried to construct as many edges as possible, so there must be a clause of the form $\square^*(l_1 \Rightarrow \boxed{a} l_2)$ such that $\eta' \models l_1$ and $\eta'' \not\models l_2$. Applying the rule $\mathbf{RES}_a^{1,1,1,0}$, we obtain the clause $\square^*(pos_{a,l_1} \Rightarrow \boxed{a} l_2)$. Observe that as $\eta' \models l_1$, then $\eta \models \neg\boxed{a}\neg l_1$; therefore, $\eta \models pos_{a,l_1}$. From this and from $\square^*(pos_{a,l_1} \Rightarrow \boxed{a} l_2)$, we have that $\eta \models \boxed{a} l_2$. Thus, the a-edge from η to η'' is removed (as $\eta'' \not\models l_2$) and the a-edge relation becomes Euclidean.

Lemma 10. *Let $\mathcal{T}$ be an unsatisfiable set of clauses in convergent systems. A contradiction can be derived by applying the resolution rules given in Section 3 and* $\mathbf{RES}_a^{1,1,1,1}$.

Proof. Assume the behaviour graph $\mathcal{G} = \langle \mathcal{N}, \mathcal{E}_1, \ldots, \mathcal{E}_n \rangle$ for $\mathcal{T}$ is not empty. If $\mathcal{T}$ is satisfiable in $\mathbf{G}_a^{1,1,1,1}$, by correspondence theory, we have that for all η in $\mathcal{G}$, $a \in \mathcal{A}$, if (η, η') and $(\eta, \eta'') \in \mathcal{E}_a$, then there exists η''' such that both (η', η''') and (η'', η''') are in $\mathcal{E}_a$. We show next that nodes that do not satisfy this condition have edges deleted from the graph and that these deletions correspond to applications of the inference rule $\mathbf{RES}_a^{1,1,1,1}$. There are two cases:

1. If there is no η''' such that $(\eta', \eta''') \in \mathcal{E}_a$, then by the graph construction there must be a clause as $\square^*(l_1 \Rightarrow \boxed{a} l_2)$ such that $\eta' \models l_1$, but $\eta''' \not\models l_2$. By applying $\mathbf{RES}_a^{1,1,1,1}$ to $\square^*(l_1 \Rightarrow \boxed{a} l_2)$, we introduce $\square^*(pos_{a,l_1} \Rightarrow \boxed{a} pos_{a,l_2})$ to the set of clauses. Now, because $\eta' \models l_1$, we have that $\eta \models pos_{a,l_1}$. By the semantics of the implication, we have that η must satisfy $\boxed{a} pos_{a,l_2}$. Now, η'' cannot satisfy pos_{a,l_2}, as if this was the case, there should be a node that would be the successor of both η' and η''. Therefore, the edge (η, η'') is removed from the graph.
2. If there is no η''' such that $(\eta'', \eta''') \in \mathcal{E}_a$, then by the graph construction there must be a clause as $\square^*(l_1 \Rightarrow \boxed{a} l_2)$ such that $\eta' \models l_1$, but $\eta''' \not\models l_2$. By applying the inference rule $\mathbf{RES}_a^{1,1,1,1}$ to $\square^*(l_1 \Rightarrow \boxed{a} l_2)$, we introduce $pos_{a,l_1} \Rightarrow \boxed{a} pos_{a,l_2}$ to the set of clauses. Reasoning as above, the edge (η, η') is removed from the graph.

Theorem 3. *Let $\mathcal{T}$ be an unsatisfiable set of clauses in $\mathbf{G}_a^{p,q,r,s}$. A contradiction can be derived by applying the resolution rules given in Section 3 and Table 2.*

Proof. By Lemmas 3, 4, 5, 6, 7, 8, 9, and 10.