Clausal Resolution for Modal Logics of Confluence – Extended Version*

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

[1] Departament of Computer Science, University of Brasília
C.P. 4466 – CEP:70.910-090 – Brasília – DF – Brazil
nalon@unb.br
[2] LoLITA and Dept. of Informatics and Applied Mathematics, UFRN, Brazil
jmarcos@dimap.ufrn.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 multimodal logics of confluence, whose Kripke semantics are based on frames characterised by appropriate instances of the Church-Rosser property. Here we restrict attention to eight families of such logics. We show how the inference rules related to the normal logics of confluence can be systematically obtained from the parametrised axioms that characterise such systems. We discuss soundness, completeness, and termination of the method. In particular, completeness can be modularly proved by showing that the conclusions of each newly added inference rule ensures that the corresponding conditions on frames hold. Some examples are given in order to illustrate the use of the method.

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 vary with the field of application to which they are tailored to apply. In the most common interpretation, formulae "$\Box p$" and "$\Diamond p$" are read as "p is necessary" and "p is possible", respectively. Evaluation of a modal formula depends upon an organised collection of scenarios known as *possible worlds*. Different modal logics assume different *accessibility relations* between such worlds. Worlds and their accessibility relations define a so-called *Kripke frame*. The evaluation of a formula hinges on such structure: given an appropriate accessibility relation and a world w, a formula $\Box p$ is satisfied at w if p is true at all worlds accessible from w; a formula $\Diamond p$ is satisfied at w if p is true at some world accessible from w.

*This is an extended version, including correctness proofs, of the paper "Clausal Resolution for Modal Logics of Confluence" accepted to IJCAR 2014.

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

‡J. Marcos was partially supported by CNPq and by the EU-FP7 Marie Curie project PIRSES-GA-2012-318986.

In normal modal logics extending the classical propositional logic, the schema $\Box(\varphi \Rightarrow \psi) \Rightarrow (\Box\varphi \Rightarrow \Box\psi)$ (the distribution axiom **K**), where φ and ψ are well-formed formulae and $\Rightarrow$ stands for classical implication, is valid, and the schematic rule $\varphi / \Box\varphi$ (the necessitation rule **Nec**) preserves validity. The weakest of these logics, named $\mathsf{K}_{(1)}$, is semantically characterised by the class of Kripke frames with no restrictions imposed on the accessibility relation. In the multimodal version, named $\mathsf{K}_{(n)}$, Kripke frames are directed multigraphs and modal operators are equipped with indexes over a set of *agents*, given by $\mathcal{A}_n = \{1, 2, \ldots, n\}$, 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 $\boxed{a}p$, with $a \in \mathcal{A}_n$, may be read as "agent a considers p to be necessary". The modal operator $\Diamond_a$ is the dual of $\boxed{a}$, being 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 its axiomatisation is given by the union of the axioms for classical propositional logic with the axiomatic schemata $\mathbf{K}_a$, namely $\boxed{a}(\varphi \Rightarrow \psi) \Rightarrow (\boxed{a}\varphi \Rightarrow \boxed{a}\psi)$, for each $a \in \mathcal{A}_n$; and the set of inference rules is given by *modus ponens* and the rule schemata $\mathbf{Nec}_a$, namely $\varphi / \boxed{a}\varphi$, for each $a \in \mathcal{A}_n$.

The basic normal multimodal logic $\mathsf{K}_{(n)}$ and its extensions have been widely used to represent and reason about complex systems. Some of the interesting extensions include the normal multimodal logics based on $\mathbf{K}_a$ and (the combination of) axioms as, for instance, $\mathbf{T}_a$ ($\boxed{a}\varphi \Rightarrow \varphi$), $\mathbf{D}_a$ ($\boxed{a}\varphi \Rightarrow \Diamond_a\varphi$), $\mathbf{4}_a$ ($\boxed{a}\varphi \Rightarrow \boxed{a}\boxed{a}\varphi$), $\mathbf{5}_a$ ($\Diamond_a\varphi \Rightarrow \boxed{a}\Diamond_a\varphi$), and $\mathbf{B}_a$ ($\Diamond_a\boxed{a}\varphi \Rightarrow \varphi$). For example, the description logic $\mathcal{ALC}$, which is employed for reasoning about ontologies, is a syntactic variant of $\mathsf{K}_{(1)}$ [22]; the epistemic logic, denoted by $\mathsf{S5}_{(n)}$, which is used in dealing with problems ranging from multi-agency to communication protocols [21, 11], can be axiomatised by combining $\mathbf{K}_a$, $\mathbf{T}_a$, and $\mathbf{5}_a$. The addition of those axioms (or their combinations) to $\mathsf{K}_{(n)}$ imposes some restrictions on the class of models where formulae are valid. Thus, a formula valid in a logic containing $\mathbf{T}_a$ is valid only if it is valid in a frame where the accessibility relation for each agent a is *reflexive*. The other axioms, $\mathbf{D}_a$, $\mathbf{4}_a$, $\mathbf{5}_a$, and $\mathbf{B}_a$, demand the accessibility relation for each agent a to be, respectively, *serial*, *transitive*, *Euclidean*, and *symmetric*.

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

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

where $a \in \mathcal{A}_n$, φ is a well-formed formula, $p, q, r, s \in \mathbb{N}$, where $\boxed{a}^0\varphi \stackrel{\text{def}}{=} \varphi$ and $\boxed{a}^{i+1}\varphi \stackrel{\text{def}}{=} \boxed{a}\boxed{a}^i\varphi$, and 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 schemata were notably studied by Lemmon [16]. Using Modal Correspondence Theory, it can be shown that the frame condition on a logic where an instance of $\mathbf{G}^{p,q,r,s}_a$ is valid corresponds to a generalised diamond-like structure representing the *Church-Rosser property* (the philosophical literature sometimes calls such property 'incestual' [8]), as illustrated in Fig. 1 [6]. To be 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 $a \in \mathcal{A}_n$. By $w\mathcal{R}^0_a w'$ we mean that $w = w'$, and $w\mathcal{R}^{i+1}_a w'$ means that there is some world w'' such that $w\mathcal{R}_a w''$ and $w''\mathcal{R}^i_a w'$. Thus, $w\mathcal{R}^i_a w'$ holds if there is an i-long $\mathcal{R}_a$-path from w to w'; alternatively, to assert that, we may also write $(w, w') \in \mathcal{R}^i_a$. Given these definitions, 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, $w_3 \in \mathcal{W}$.

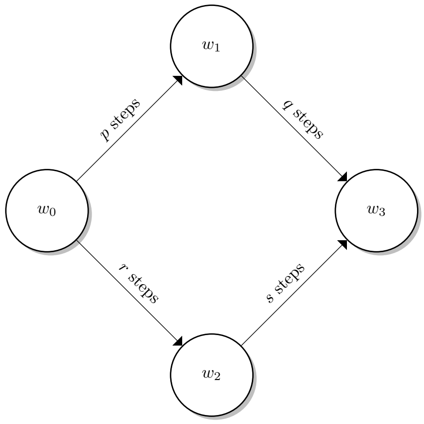

Fig. 1. Church-Rosser property for frames where $\mathbf{G}_a^{p,q,r,s} = \langle a \rangle^p \boxed{a}^q \varphi \Rightarrow \boxed{a}^r \langle a \rangle^s \varphi$ is valid.

Many well-known modal axiomatic systems are identified with particular logics of confluence. For instance, $\mathsf{T}_{(n)}$ corresponds to $\mathsf{K}_{(n)}^{0,1,0,0}$, namely a normal modal logic in which the axiom $\boxed{a}\varphi \Rightarrow \varphi$ is valid, for all $a \in \mathcal{A}_n$ and any formula φ. The axiom $\mathbf{4}_a$ may be written as $\mathbf{G}_a^{0,1,2,0}$, that is, $\boxed{a}^1 \varphi \Rightarrow \boxed{a}^2 \varphi$. The Geach axiom $\mathbf{G1}_a$ is given by $\mathbf{G}_a^{1,1,1,1}$ ($\langle a \rangle \boxed{a} \varphi \Rightarrow \boxed{a} \langle a \rangle \varphi$). Formulae in $\mathsf{K}_{(n)}^{1,1,1,1}$ are satisfiable if, and only if, they are satisfiable in a model with n relations satisfying the so-called 'diamond property', and analogous claims hold for instance concerning formulae of $\mathsf{T}_{(n)}$ and models whose relations are all reflexive, and formulae of $\mathsf{4}_{(n)}$ and models whose relations are all transitive.

Logics of confluence are interesting not only because they encompass a great number of normal modal logics as particular examples, but also in view of their attractive computational behaviour. Indeed, if we think of multimodal frames as abstract rewriting systems, for instance, and think of modal languages as a way of obtaining an internal and local perspective on such frames, then each given notion of confluence ensures that certain different paths of transformation will eventually lead to the same result. Having a decidable proof procedure for a logic underlying such class of frames helps in establishing a direct form of verifying the properties of the structures that they represent.

As a contribution towards a uniform approach to the development of proof methods for logics of confluence, in this work we deal with the logics where $p, q, r, s \in \{0, 1\}$. Table 1 shows the relevant axiomatic schemata, some standard names by which they are known, and the corresponding conditions on frames. The axiom $\mathbf{G}_a^{0,1,1,1}$ seems not to be named in the literature; the corresponding property follows the naming convention given in [5, pg. 127]. 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 obvious instances of classical tautologies and are thus not included in Table 1. Also, given the duality between $\boxed{a}$ and $\Diamond_a$, $\mathbf{G}_a^{p,q,r,s}$ is semantically equivalent to $\mathbf{G}_a^{r,s,p,q}$. Thus, there are in fact eight families of multimodal logics related to the axioms $\mathbf{G}_a^{p,q,r,s}$, where $p, q, r, s \in \{0, 1\}$.

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

Table 1. Axioms and corresponding conditions on frames.

We present a clausal resolution-based method for solving the satisfiability problem in logics axiomatised by $\mathbf{K}_a$ plus $\mathbf{G}_a^{p,q,r,s}$, where $p, q, r, s \in \{0, 1\}$. The resolution calculus is based on that of [18], 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 discuss soundness, completeness, and termination. Full proofs can be found in Appendix A.

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

The set $\mathsf{WFF}_{\mathsf{K}_{(n)}}$ of *well-formed formulae* of the logic $\mathsf{K}_{(n)}$ is constructed from a denumerable set of *propositional symbols*, $\mathcal{P} = \{p, q, p', q', p_1, q_1, \ldots\}$, the negation symbol $\neg$, the conjunction symbol $\wedge$, the propositional constant **true**, and a unary connective $\boxed{a}$ for each agent a in the finite set of agents $\mathcal{A}_n = \{1, \ldots, n\}$. When $n = 1$, we often omit the index, that is, $\Box\varphi$ stands for $\boxed{1}\varphi$. As usual, $\Diamond_a$ is introduced as an abbreviation for $\neg\boxed{a}\neg$. A *literal* is either a propositional symbol 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, $\neg l$ denotes $\neg p$ if l is the propositional symbol p, and $\neg l$ denotes 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 $a \in \mathcal{A}_n$.

We present the semantics of $\mathsf{K}_{(n)}$, as usual, in terms of Kripke frames.

Definition 1. *A* Kripke frame $\mathcal{S}$ for n agents over $\mathcal{P}$ *is given by 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 frame* $\mathcal{S}$ *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 propositional symbols.*

The so-called accessibility relation $\mathcal{R}_a$ is a binary relation that captures the notion of relative possibility from the viewpoint of agent a: A pair (w, w') is in $\mathcal{R}_a$ if agent a considers world w' possible, given the information available to her in world w. We write $\langle \mathcal{M}, w \rangle \models \varphi$ (resp. $\langle \mathcal{M}, w \rangle \not\models \varphi$) to say that φ is satisfied (resp. not satisfied) at the world w in the Kripke model $\mathcal{M}$.

Definition 2. *Satisfaction of a formula at a given world* w *of a model* $\mathcal{M}$ *is set by:*

- $\langle \mathcal{M}, w \rangle \models$ **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. A formula φ is said to be *satisfied in the model* $\mathcal{M} = (\mathcal{S}, \pi)$ *of the Kripke frame* $\mathcal{S} = (\mathcal{W}, w_0, \mathcal{R}_1, \ldots, \mathcal{R}_n)$ if $\langle \mathcal{M}, w_0 \rangle \models \varphi$; the formula φ is *satisfiable in a Kripke frame* $\mathcal{S}$ 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* $\mathcal{C}$ *of Kripke frames* if it is satisfied in any model of any Kripke frame belonging to the class $\mathcal{C}$.

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

In [18], a sound, complete, and terminating resolution-based method for $\mathsf{K}_{(n)}$, which in this paper we call RES_K, is introduced. As the proof-method for logics of confluence presented here relies on RES_K, in order to keep the present paper self-contained, we reproduce the corresponding inference rules here and refer the reader to [18] for a detailed account of the method. The approach taken in the resolution-based method for $\mathsf{K}_{(n)}$ is clausal: a formula to be tested for (un)satisfiability is first 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). 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 $\mathcal{M} = (\mathcal{W}, w_0, \mathcal{R}_1, \ldots, \mathcal{R}_n, \pi)$, we have that $\langle \mathcal{M}, w \rangle \models \textbf{start}$ if, and only if, $w = w_0$. A formula in SNF is represented by a conjunction of clauses, which are true at all reachable states, that is, they have the general form $\bigwedge_i \Box^* A_i$, where A_i is a clause and $\Box^*$, the universal operator, is characterised by (the greatest fixed point of) $\Box^*\varphi \Leftrightarrow \varphi \wedge \bigwedge_{a \in \mathcal{A}_n} \boxed{a}\Box^*\varphi$, for a formula φ. Observe that satisfaction of $\Box^*\varphi$ imposes that φ must hold at the actual world w and at every world reachable from w, where reachability is defined in the usual (graph-theoretic) way. Clauses have one of the following forms:

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

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

The translation to SNF uses rewriting of classical operators and the renaming technique [20], 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. Given a formula φ, the translation procedure is applied to $\Box^*(\textbf{start} \Rightarrow t_0) \wedge \Box^*(t_0 \Rightarrow \varphi)$, where t_0 is a new propositional symbol. The universal operator, which surrounds all clauses, ensures that the clauses generated by the translation of a formula are true at all reachable worlds. Classical rewriting is used to remove some classical operators from φ (e.g. $\Box^*(t \Rightarrow \psi_1 \wedge \psi_2)$ is rewritten as $\Box^*(t \Rightarrow \psi_1) \wedge \Box^*(t \Rightarrow \psi_2)$). Renaming is used to replace complex subformulae in disjunctions (e.g. if ψ_2 is not a literal, $\Box^*(t \Rightarrow \psi_1 \vee \psi_2)$ is rewritten as $\Box^*(t \Rightarrow \psi_1 \vee t_1) \wedge \Box^*(t_1 \Rightarrow \psi_2)$, where t_1 is a new propositional symbol) or in the scope of modal operators (e.g. if ψ is not a literal, $\Box^*(t \Rightarrow \boxed{a}\psi)$ is rewritten as $\Box^*(t \Rightarrow \boxed{a}t_1) \wedge \Box^*(t_1 \Rightarrow \psi)$, where t_1 is a new propositional symbol). We refer the reader to [18] for details on the transformation rules that define the translation to SNF, their correctness, and examples of their application.

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 classical propositional fragment of the combined logic. An initial clause may be resolved with either a literal clause or another initial clause (rules **IRES1** and **IRES2**). Literal clauses may be resolved together (**LRES**).

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

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 may resolve two or more $\boxed{a}$-clauses (rules **MRES** and **NEC2**); or several $\boxed{a}$-clauses and a literal clause (rules **NEC1** and **NEC3**). The modal inference rules are:

$$
\textbf{[MRES]}\ \begin{array}{c} \Box^*(l_1 \Rightarrow \boxed{a}\, l) \\ \Box^*(l_2 \Rightarrow \neg \boxed{a}\, l) \\ \hline \Box^*(\mathbf{true} \Rightarrow \neg l_1 \lor \neg l_2) \end{array}
\qquad
\textbf{[NEC2]}\ \begin{array}{c} \Box^*(l'_1 \Rightarrow \boxed{a}\, l_1) \\ \Box^*(l'_2 \Rightarrow \boxed{a} \neg l_1) \\ \Box^*(l'_3 \Rightarrow \neg \boxed{a}\, l_2) \\ \hline \Box^*(\mathbf{true} \Rightarrow \neg l'_1 \lor \neg l'_2 \lor \neg l'_3) \end{array}
$$

$$
\textbf{[NEC1]}\ \begin{array}{c} \Box^*(l'_1 \Rightarrow \boxed{a} \neg l_1) \\ \vdots \\ \Box^*(l'_m \Rightarrow \boxed{a} \neg l_m) \\ \Box^*(l' \Rightarrow \neg \boxed{a}\, l) \\ \Box^*(\mathbf{true} \Rightarrow l_1 \lor \ldots \lor l_m \lor l) \\ \hline \Box^*(\mathbf{true} \Rightarrow \neg l'_1 \lor \ldots \lor \neg l'_m \lor \neg l') \end{array}
\qquad
\textbf{[NEC3]}\ \begin{array}{c} \Box^*(l'_1 \Rightarrow \boxed{a} \neg l_1) \\ \vdots \\ \Box^*(l'_m \Rightarrow \boxed{a} \neg l_m) \\ \Box^*(l' \Rightarrow \neg \boxed{a}\, l) \\ \Box^*(\mathbf{true} \Rightarrow l_1 \lor \ldots \lor l_m) \\ \hline \Box^*(\mathbf{true} \Rightarrow \neg l'_1 \lor \ldots \lor \neg l'_m \lor \neg l') \end{array}
$$

The rule **MRES** is a syntactic variation of classical resolution, as a formula and its negation cannot be true at the same state. The rule **NEC1** corresponds to necessitation (applied to $(\neg l_1 \land \ldots \land \neg l_m \Rightarrow \neg l)$, which is equivalent to the literal clause in the premises) and several applications of classical resolution. The rule **NEC2** is a special case of **NEC1**, as the parent clauses can be resolved with the tautology **true** $\Rightarrow l_1 \lor \neg l_1 \lor l_2$. The rule **NEC3** is similar to **NEC1**, 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 **NEC1**–**NEC3** 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 [18]. Note that for **NEC1**, we may have $m = 0$; for **NEC2** the number of premises is fixed; and that for **NEC3**, if $m = 0$, then the literal clause in the premises is given by **true** $\Rightarrow$ **false**, which cannot be satisfied in any model. Therefore, **NEC3** is not applied when $m = 0$.

We define a derivation as a sequence of sets of clauses $\mathcal{T}_0, \mathcal{T}_1, \ldots$, where $\mathcal{T}_i$ results from adding to $\mathcal{T}_{i-1}$ the resolvent obtained by an application of an inference rule of $\mathsf{RES_K}$ to clauses in $\mathcal{T}_{i-1}$. A derivation *terminates* if, and only if, either a contradiction, in the form of $\Box^*(\mathbf{start} \Rightarrow \mathbf{false})$ or $\Box^*(\mathbf{true} \Rightarrow \mathbf{false})$, is derived or no new clauses can be derived by further application of the resolution rules of $\mathsf{RES_K}$. 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 1. We wish to check whether the formula $\boxed{1}\,\boxed{2}(a \wedge b) \Rightarrow \boxed{1}(\boxed{2}a \wedge \boxed{2}b)$ is valid in $\mathsf{K}_{(2)}$. The translation of its negation into the normal form is given by clauses (1)–(9) below. Then the inference rules are applied until **false** is generated. In order to improve readability, the universal operator is suppressed. The full refutation follows.

1. $\textbf{start} \Rightarrow t_1$
2. $t_1 \Rightarrow \boxed{1}\,t_2$
3. $t_2 \Rightarrow \boxed{2}\,t_3$
4. $\textbf{true} \Rightarrow \neg t_3 \vee a$
5. $\textbf{true} \Rightarrow \neg t_3 \vee b$
6. $t_1 \Rightarrow \neg\,\boxed{1}\neg t_4$
7. $\textbf{true} \Rightarrow \neg t_4 \vee t_5 \vee t_6$
8. $t_5 \Rightarrow \neg\,\boxed{2}\,a$
9. $t_6 \Rightarrow \neg\,\boxed{2}\,b$
10. $\textbf{true} \Rightarrow \neg t_2 \vee \neg t_5$ $[\mathbf{NEC1}, 3, 8, 4]$
11. $\textbf{true} \Rightarrow \neg t_2 \vee \neg t_4 \vee t_6$ $[\mathbf{LRES}, 10, 7]$
12. $\textbf{true} \Rightarrow \neg t_2 \vee \neg t_6$ $[\mathbf{NEC1}, 3, 9, 5]$
13. $\textbf{true} \Rightarrow \neg t_2 \vee \neg t_4$ $[\mathbf{LRES}, 12, 11]$
14. $\textbf{true} \Rightarrow \neg t_1$ $[\mathbf{NEC1}, 2, 6, 13]$
15. $\textbf{start} \Rightarrow \textbf{false}$ $[\mathbf{IRES1}, 14, 1]$

Clauses (10) and (12) are obtained by applications of **NEC1** to clauses in the context of agent 2. Clause (14) is obtained by an application of the same rule, but in the context of agent 1. Clauses (11) and (13) result from applications of resolution to the propositional part of the language shared by both agents. Clause (15) 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 $\mathsf{RES_K}$, 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 $\langle a\rangle^{p}\,\boxed{a}^{q}\psi$ holds, we add some new clauses that ensure that $\boxed{a}^{r}\langle a\rangle^{s}\psi$ also holds. If this is not the case, that is, if the set of clauses implies that $\neg\,\boxed{a}^{r}\langle a\rangle^{s}\psi$ holds, then a contradiction is found by applying the inference rules for $\mathsf{K}_{(n)}$. 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).

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

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

Soundness is checked by showing that the transformation of a formula $\varphi \in \mathsf{WFF}_{\mathsf{K}_{(n)}}$ into its normal form is satisfiability-preserving and that the application of the inference rules are also satisfiability-preserving. Satisfiability-preserving results for the transformation into SNF are provided in [18]. To extend the soundness results so as to cover the new inference rules, note that the conclusions of the inference rules in Table 2 are derived using the semantics of the universal operator and the distribution axiom, $\mathbf{K}_a$. For $\mathbf{RES}_a^{p,1,r,s}$, we have that the premise $\Box^*(l \Rightarrow \boxed{a} l')$ 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 $\boxed{a}^r \langle a \rangle^s l'$ in $\mathsf{K}_{(n)}^{p,1,r,s}$, by classical reasoning, we have that $\Box^*(\neg \boxed{a}^p \neg l \Rightarrow \neg \boxed{a}^p \neg \boxed{a} l')$ implies $\Box^*(\langle a \rangle^p l \Rightarrow \boxed{a}^r \langle a \rangle^s l')$, the conclusion of $\mathbf{RES}_a^{p,1,r,s}$. Soundness of the inference rule $\mathbf{RES}_a^{p,0,r,s}$ can be proved in a similar way.

As the conclusions of the above inference rules may contain complex formulae, they might need to be rewritten into the normal form. Thus, 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 $\tau(\varphi)$ be the set of clauses resulting from the translation of φ into the normal form. Let $\mathcal{L}(\tau(\varphi))$ be the set of literals that might occur in the clause set, that is, for all $p \in \mathcal{P}$ such that p occurs in $\tau(\varphi)$, we have that both p and $\neg p$ are in $\mathcal{L}(\tau(\varphi))$. The set of *definition clauses* is given by

$$\Box^*(pos_{a,l} \Rightarrow \neg \boxed{a} \neg l)$$
$$\Box^*(\neg pos_{a,l} \Rightarrow \boxed{a} \neg l)$$

for all $l \in \mathcal{L}(\tau(\varphi))$, 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 $pos_{a,l} \Leftrightarrow \neg \boxed{a} \neg l$. Note that we have definition clauses for every propositional symbol and its negation, e.g. for a propositional symbol $p \in \tau(\varphi)$, we have the definition clauses $\Box^*(pos_{a,p} \Rightarrow \neg \boxed{a} \neg p)$, $\Box^*(\neg pos_{a,p} \Rightarrow \boxed{a} \neg p)$, $\Box^*(pos_{a,\neg p} \Rightarrow \neg \boxed{a} p)$, and $\Box^*(\neg pos_{a,\neg p} \Rightarrow \boxed{a} p)$, for every $a \in \mathcal{A}_n$ occurring in $\tau(\varphi)$. We assume the set of definition clauses to be available whenever those symbols are used. It is also important to note that those new definition symbols and the respective definition clauses can all be introduced at the beginning of the application of the resolution method because we do not need definition clauses applied to definition symbols in the proofs, as given in the completeness proof (see Appendix A). 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.

As discussed above and from the results in [18], we can establish the soundness of the proof method.

Theorem 1. *1 The resolution-based calculi for logics of confluence are sound.*

Proof (Sketch). The transformation into the normal form is satisfiability preserving [18]. Given a set $\mathcal{T}$ of clauses and a model $\mathcal{M}$ that satisfies $\mathcal{T}$, we can construct a model $\mathcal{M}'$ for the union of $\mathcal{T}$ and the definition clauses, where $\mathcal{M}$ and $\mathcal{M}'$ may differ only in the valuation of the definition symbols. By setting properly the valuations in $\mathcal{M}'$, we have that $\langle \mathcal{M}', w \rangle \models pos_{a,p}$ if and only if $\langle \mathcal{M}, w \rangle \models \langle a \rangle p$, for any $w \in \mathcal{W}$. Soundness of the inference rules for $\mathsf{RES_K}$ is also given in [18]. Soundness of $\mathbf{RES}_a^{p,1,r,s}$ and $\mathbf{RES}_a^{p,0,r,s}$ follow from the axiomatisation of $\mathsf{K}_{(n)}^{p,q,r,s}$.

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

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

Table 3. Inference Rules for several instances of $\mathbf{G}_a^{p,q,r,s}$

Table 3 shows the inference rules for each specific instance of $\mathbf{G}_a^{p,q,r,s}$, where $p, q, r, s \in \{0, 1\}$, 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 inference rules in Table 3 are obtained directly from Table 2. For instance, the rule for reflexive systems, i.e. where the axiom $\mathbf{G}_a^{0,1,0,0}$ is valid, has the form $\Box^*(l \Rightarrow \boxed{a} l')/\Box^*(\langle a \rangle^0 l \Rightarrow \boxed{a}^0 \langle a \rangle^0 l')$ in Table 2; in Table 3, the conclusion is rewritten in its normal form, that is, $\Box^*(\mathbf{true} \Rightarrow \neg l \vee l')$. For other systems, the form of the inference rules are 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, those systems where the axiom $\mathbf{G}_a^{1,1,0,0}$ is valid. From Table 2, in symmetric systems, for a premise of the form $\Box^*(l \Rightarrow \boxed{a} l')$, the conclusion is given by $\Box^*(\langle a \rangle l \Rightarrow l')$, which is translated into the normal form as $\Box^*(\mathbf{true} \Rightarrow \neg pos_{a,l} \vee l')$. We have chosen, however, to translate the conclusion as $\Box^*(\neg l' \Rightarrow \boxed{a} \neg l)$, which is semantically equivalent to the conclusion obtained by the general inference rule, but avoids the use of definition 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 do not always 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 for instance that the inference rules which introduce modalities in their conclusions from literal clauses (that is, the inference rules $\mathbf{RES}_a^{0,0,r,s}$) are not needed for completeness. We also show that we need just one specific inference rule for logics in which $\mathbf{G}_a^{0,1,1,1}$ and $\mathbf{5}_a$ are valid: $\mathbf{RES}_a^{0,1,1,1}$ and $\mathbf{RES}_a^{1,0,1,1}$, respectively.

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, and the inference rule $\mathbf{RES}_a^{p,q,r,s}$ are applied to $\tau(\varphi)$ and the set of definition clauses. The extra inference rules for $\mathsf{K}_{(n)}^{p,q,r,s}$ do not need to be applied to clauses if such application generates new nested definition symbols, that is, we do not need definition clauses for definition symbols. For instance, the application of $\mathbf{RES}_a^{1,1,1,1}$ to a clause of the form $\Box^*(l \Rightarrow \boxed{a} pos_{a,l'})$ would result in $\Box^*(pos_{a,l} \Rightarrow \boxed{a} pos_{a,pos_{a,l'}})$. Although it is not incorrect to apply the inference rules to such a clause, this might cause the method not to terminate. We can show, however, that the application of inference rules to clauses which would result in nested literals is not needed for completeness, as the restrictions imposed by those symbols are already ensured by existing definition symbols and relevant inference rules (see Theorem 3 below). This ensures that no new definition symbols are introduced by the proof method.

Completeness is proved by showing that, for each specific logic of confluence, 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. 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 check completeness for proof methods in modal logics (see [3], for instance, for definitions and examples). Intuitively, nodes in the graph correspond to states and are defined as maximally consistent sets of literals and modal literals occurring in the set of clauses, including those literals introduced by definition clauses. That is, for any literal l occurring in the set of clauses, including definition clauses, and agents $a \in \mathcal{A}_n$, 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. Edges or nodes that do not satisfy the set of clauses are deleted from the graph. Such deletions correspond to applications of one or more of the inference rules. 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 for RES_K, given in Section 3, and the inference rules for the specific logic of confluence, presented in Table 3.

Theorem 2. *2 Let $\mathcal{T}$ be an unsatisfiable set of clauses in $\mathbf{G}_a^{p,q,r,s}$, with $p, q, r, s \in \{0, 1\}$. A contradiction can be derived by applying the resolution rules for* RES_K*, presented in Section 3, and Table 3.*

Proof (Sketch). We construct a behaviour graph and show that the application of rules in Table 3 removes nodes and edges where the corresponding frame condition does not hold. The full proof is provided in the Appendix A.

Theorem 3. *3 The resolution-based calculi for logics of confluence terminate.*

Proof (Sketch). From the completeness proof, the introduction of a literal such as $pos_{a,pos_{a,l}}$ for an agent a and literal l is not needed. We can show that the restrictions imposed by such clauses, together with the resolution rules for each specific logical system, are enough to ensure that the corresponding frame condition already holds. As the proof method does not introduce new literals in the clause set, there is only a finite number of clauses that can be expressed. Therefore, the proof method is terminating.

Example 2. We show that $\varphi \stackrel{\text{def}}{=} p \Rightarrow \boxed{1}\Diamond_1 p$, which is an instance of $\mathbf{B}_1$, is a valid formula in symmetric systems. As symmetry is implied by reflexivity and Euclideanness, instead of using $\mathbf{RES}_1^{1,1,0,0}$, we combine the inference rules for both $\mathbf{T}_1$ and $\mathbf{5}_1$. Clauses (1)–(4) correspond to the translation of the negation of φ into the normal form. Clauses (5)–(8) are the definition clauses used in the proof.

1.	$\mathbf{start} \Rightarrow t_0$		9.	$\mathbf{true} \Rightarrow \neg t_0 \vee pos_{1,t_1}$	$[\mathbf{MRES}, 5, 3]$
2.	$\mathbf{true} \Rightarrow \neg t_0 \vee p$		10.	$\mathbf{true} \Rightarrow \neg t_1 \vee \neg pos_{1,p}$	$[\mathbf{MRES}, 7, 4]$
3.	$t_0 \Rightarrow \neg \boxed{1} \neg t_1$		11.	$pos_{1,p} \Rightarrow \boxed{1} pos_{1,p}$	$[\mathbf{RES}_1^{1,0,1,1}, 7]$
4.	$t_1 \Rightarrow \boxed{1} \neg p$		12.	$\mathbf{true} \Rightarrow \neg pos_{1,p} \vee \neg pos_{1,t_1}$	$[\mathbf{NEC1}, 11, 6, 10]$
5.	$\neg pos_{1,t_1} \Rightarrow \boxed{1} \neg t_1$	$[\mathbf{Def}.\ pos_{1,t_1}]$	13.	$\mathbf{true} \Rightarrow \neg p \vee pos_{1,p}$	$[\mathbf{RES}_1^{0,1,0,0}, 8]$
6.	$pos_{1,t_1} \Rightarrow \neg \boxed{1} \neg t_1$	$[\mathbf{Def}.\ pos_{1,t_1}]$	14.	$\mathbf{true} \Rightarrow \neg p \vee \neg pos_{1,t_1}$	$[\mathbf{LRES}, 13, 12]$
7.	$pos_{1,p} \Rightarrow \neg \boxed{1} \neg p$	$[\mathbf{Def}.\ pos_{1,p}]$	15.	$\mathbf{true} \Rightarrow \neg t_0 \vee \neg p$	$[\mathbf{LRES}, 14, 9]$
8.	$\neg pos_{1,p} \Rightarrow \boxed{1} \neg p$	$[\mathbf{Def}.\ pos_{1,p}]$	16.	$\mathbf{true} \Rightarrow \neg t_0$	$[\mathbf{LRES}, 15, 2]$
			17.	$\mathbf{start} \Rightarrow \mathbf{false}$	$[\mathbf{IRES1}, 16, 1]$

Clause (11) results from applying the Euclidean inference rule to clause (7). Clause (13) results from applying the reflexive inference rule to (8). The remaining clauses are derived by the resolution calculus for $\mathsf{K}_{(1)}$. As a contradiction is found, given by clause (17), 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 multimodal systems where axioms of the form

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

where $p, q, r, s \in \{0, 1\}$, 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.

We have proved completeness of the proof method presented in this paper for eight families of logics and their fusions. The inference rules for particular instances of these logics can be systematically obtained and the resulting calculus can be implemented by adding to the existing prover for $\mathsf{K}_{(n)}$ [24] the clauses dependent on the clause-set. Efficiency, of course, depends on several aspects. Firstly, for certain classes of problems, dedicated proof methods might be more efficient. For instance, if the satisfiability problem for a particular logic is in NP (as in the case of $\mathsf{S5}_{(1)}$), then our procedure may be less efficient as the satisfiability problem for $\mathsf{K}_{(1)}$ is already PSPACE-complete [15]. Secondly, efficiency might depend on the inference rules chosen to produce proofs for a specific logic. For instance, for $\mathsf{S5}_{(n)}$, the user can choose the inference rules related to reflexivity and Euclideanness, or choose the inference rules related to seriality, symmetry, and Euclideanness. The number of inference rules used to test the unsatisfiability of a set of clauses for a particular logic might affect the number of clauses generated by the resolution method as well as the size of the proof. As in the case of derived inference rules in other proof methods, using more inference rules might lead to shorter proofs. Thirdly, as in the case of the resolution-based method for propositional logic, efficiency might be affected by strategies used to search for a proof. Future work includes the design of strategies for RES_K(n) and for specific logics of confluence. Fourthly, efficiency might also depend on the form of the input problem. For instance, comparisons between tableaux methods and resolution methods [14, 13] have shown that there is no overall better approach: for some problems resolution proof methods behave better, for others tableaux based methods behave better. Providing a resolution-based method for the logics axiomatised by $\mathbf{K}_a$ and $\mathbf{G}_a^{p,q,r,s}$ gives the user a choice for automated tools that can be used depending on the type of the input formulae.

There are quite a few dedicated methods for the logics presented in this paper. In general, however, those methods do not provide a systematic way of dealing with logics based on similar axioms or their extensions. Therefore, we restrict attention here to methods related to logics of confluence. Tableaux methods for logics of confluence where the mono-modal axioms **T**, **D**, **B**, **4**, **5**, **De** (for density, the converse of **4**), and **G** are valid, can be found in [7, 9]. For each of those axioms, a tableau inference rule is given. The inference rules can then be combined in order to provide proof methods for modal logics under $\mathsf{S5}_{(1)}$. Whilst the tableaux procedures in [7, 9] are designed for mono-modal logics they seem to be extendable to multimodal logics as long as there are no interactions between modalities. Those procedures do not cover all the logics investigated in this paper. In [2], labelled tableaux are given for the mono-modal logics axiomatised by **K** and axioms $\mathbf{G}^{p,q,r,s}$ where $q = s = 0$ implies $p = r = 0$. This restriction avoids the introduction of inference rules related to the identity predicate, but also excludes, for instance, functional and modally banal systems, which are treated by the method introduced in the present paper. In [4], hybrid logic tableaux methods for logics of confluence are given: the inference rules create nodes, labelled by nominals. The nominals are used in order to eliminate the Skolem function related to the existential quantifier in the first-order sentence corresponding to the axiom $\mathbf{G}_a^{p,q,r,s}$. This proof method provides tableau rules for all instances of the axiom. Soundness and completeness are discussed, but termination of the method is not dealt with and it is not clear what are the bounds for creating new nodes in the general case. In [12], sound, complete, and terminating display calculi for tense logics and some of its extensions, including those with the axiom $\mathbf{G}_a^{p,q,r,s}$, are presented. It has been shown that these calculi have the property of *separation*, that is, they provide complete proof methods for the component fragments. The paper investigates the relation between the display calculi and deep inference systems (where the sequent rules can be applied at any node of a proof tree). By finding appropriate propagation rules for the fusion of tense logic with either $\mathsf{S4}_{(1)}$, $\mathsf{S5}_{(1)}$, or functional systems, completeness of search strategies are presented. However, propagation rules for the axiom of convergence, $\mathbf{G1}$, or for the combination of path axioms (i.e. axioms of the form $\Diamond^i \varphi \Rightarrow \Diamond^j \varphi$) with seriality are not given. Also related, in [1], prefixed tableaux procedures for confluence logics that validate the multimodal version of the axiom $\Diamond_{a} \boxed{b} \varphi \Rightarrow \boxed{c} \Diamond_{d} \varphi$, where φ is a formula, are given. Note that the logics in [1] are systems with instances of the axiom $\mathbf{G}^{1,1,1,1}_{a,b,c,d}$, that is, a logic which allows the interaction of the agents $a, b, c, d \in \mathcal{A}_n$, and might lead to undecidable systems.

To the best of our knowledge, there are no resolution-based proof methods for logics of confluence. However, resolution-based methods for modal logics, based on translation into first-order logic, have been proposed for several modal logics. A survey on translation-based approaches for non-transitive modal logics (i.e. modal logics that do not include the axiom **4**) can be found in [19]. The translation-based approach has the clear advantage of being easily implemented, making use of well-established theorem-provers, and dealing with any logic that can be embedded into first-order, should it be decidable or not. However, first-order provers cannot deal easily with logics that embed some properties which are covered by particular axioms of confluence (e.g. functionality). In order to avoid such problematic fragments within first-order logic, the axiomatic translation principle for modal logic, introduced in [23], besides using the standard translation of a modal formulae into first-order, takes an axiomatisation for a particular modal logic and introduces a set of first-order modal axioms in the form of *schema clauses*. As an example, adapted from [23], in order to prove that $\boxed{a} \neg \boxed{a} p$ is satisfiable in $\mathsf{KT4}_{(n)}$, for each modal subformula (i.e. $\boxed{a} \neg \boxed{a} p$ and $\boxed{a} p$) and for each considered axiom (i.e. **T** and **4**), one schema clause is added, resulting in:

$$
\begin{array}{cc}
\neg Q_{\boxed{a} \neg \boxed{a} p}(x) \vee \neg R(x,y) \vee Q_{\boxed{a} \neg \boxed{a} p}(y) & \qquad \neg Q_{\boxed{a} \neg \boxed{a} p}(x) \vee Q_{\neg \boxed{a} p}(y) \\
\neg Q_{\boxed{a} p}(x) \vee \neg R(x,y) \vee Q_{\boxed{a} p}(y) & \qquad \neg Q_{\boxed{a} p}(x) \vee Q_p(y)
\end{array}
$$

where the predicate $Q_\varphi(x)$ can be read as φ holds at world x and R is the predicate symbol to express the accessibility relation for agent a. Note that the clauses on the left are related to transitivity (**4**) and the two clauses on the right are related to reflexivity (**T**). The axiomatic translation approach is similar to the approach taken in the present paper and in [18] as the schema clauses provide a way of talking about properties of the accessibility relation. As in our case, soundness follows easily from the properties of the translation. Termination follows from the fact that only a finite number of schema clauses are needed. However, as in the case of the proof method presented here, general completeness of the method is difficult to be proved and it is given only for particular families of logics. In [10], a translation-based approach for properties which can be expressed by regular grammar logics (including transitivity and Euclideaness) is given. Completeness for the general method is also difficult and it has been proved for some families of logics.

In the present 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, by adding inference rules of the following form:

$$[\mathbf{RES}_a^{p,q,r,s}]\ \frac{\Box^*(l \Rightarrow \Diamond_a^{\,p} \Box_a^{\,r} l')}{\Box^*(l \Rightarrow \Box_a^{\,r} \Diamond_a^{\,s} l')}$$

which requires search for clauses that correspond to the normal form of the premise and the introduction of as many new definition symbols as the number of modalities occurring in the conclusion. The inference rule $\mathbf{RES}_a^{p,q,r,s}$ is obviously sound, but we have yet to identify the restrictions on the number of new propositional symbols introduced by the method in order to ensure termination. Future work includes this extension, the complexity analysis, the implementation of the proof method, and practical comparisons with other methods.

References

1. M. Baldoni, L. Giordano, and A. Martelli. A tableau calculus for multimodal logics and some (un)decidability results. In *Proc. of TABLEAUX-98*, pages 44–59. Springer-Verlag, 1998.
2. D. Basin, S. Matthews, and L. Viganò. Labelled propositional modal logics: Theory and practice. *J. Log. Comput*, 7(6):685–717, 1997.
3. P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, Cambridge, England, 2001.
4. P. Blackburn, E. L. E. Dialogue, and B. T. Cate. Beyond pure axioms: Node creating rules in hybrid tableaux. In C. E. Areces, P. Blackburn, M. Marx, and U. Sattler, editors, *Hybrid Logics*, pages 21–35, July 25 2002.
5. G. S. Boolos. *The Logic of Provability*. Cambridge University Press, 1993.
6. W. A. Carnielli and C. Pizzi. *Modalities and Multimodalities*, volume 12 of *Logic, Epistemology, and the Unity of Science*. Springer, 2008.
7. M. A. Castilho, L. F. del Cerro, O. Gasquet, and A. Herzig. Modal tableaux with propagation rules and structural rules. *Fundamenta Informaticae*, 32(3-4):281–297, 1997.
8. B. Chellas. *Modal Logic : An Introduction*. Cambridge University Press, 1980.
9. L. F. del Cerro and O. Gasquet. Tableaux based decision procedures for modal logics of confluence and density. *Fundamenta Informaticae*, 40(4):317–333, 1999.
10. S. Demri and H. Nivelle. Deciding regular grammar logics with converse through first-order logic. *Journal of Logic, Language and Information*, 14(3):289–329, 2005.
11. R. Fagin, J. Halpern, Y. Moses, and M. Vardi. *Reasoning About Knowledge*. MIT Press, 1995.
12. R. Goré, L. Postniece, and A. Tiu. On the correspondence between display postulates and deep inference in nested sequent calculi for tense logics. *Logical Methods in Computer Science*, 7(2), 2011.
13. R. Goré, J. Thomson, and F. Widmann. An experimental comparison of theorem provers for CTL. In C. Combi, M. Leucker, and F. Wolter, editors, *TIME 2011, Lübeck , Germany, September 12-14, 2011*, pages 49–56. IEEE, 2011.
14. U. Hustadt and R. A. Schmidt. Scientific benchmarking with temporal logic decision procedures. In D. Fensel, F. Giunchiglia, D. McGuinness, and M.-A. Williams, editors, *Proceedings of the KR'2002*, pages 533–544. Morgan Kaufmann, 2002.
15. R. E. Ladner. The computational complexity of provability in systems of modal propositional logic. *SIAM J. Comput.*, 6(3):467–480, 1977.
16. E. J. Lemmon and D. Scott. *The Lemmon Notes: An Introduction to Modal Logic*. Edited by Segerberg, K., Basil Blackwell, 1977.

17. C. Nalon and C. Dixon. Anti-prenexing and prenexing for modal logics. In *Proceedings of the 10th European Conference on Logics in Artificial Intelligence*, Liverpool, UK, 2006.
18. C. Nalon and C. Dixon. Clausal resolution for normal modal logics. *J. Algorithms*, 62:117–134, July 2007.
19. H. D. Nivelle, R. A. Schmidt, and U. Hustadt. Resolution-based methods for modal logics. *Logic J. IGPL*, 8:2000, 2000.
20. D. A. Plaisted and S. A. Greenbaum. A Structure-Preserving Clause Form Translation. *Journal of Logic and Computation*, 2:293–304, 1986.
21. A. Rao and M. Georgeff. Modeling Rational Agents within a BDI-Architecture. In R. Fikes and E. Sandewall, editors, *Proceedings of KR&R-91*, pages 473–484. Morgan-Kaufmann, Apr. 1991.
22. K. Schild. A Correspondence Theory for Terminological Logics. In *Proceedings of the 12th IJCAI*, pages 466–471, 1991.
23. R. A. Schmidt and U. Hustadt. The axiomatic translation principle for modal logic. *ACM Transactions on Computational Logic*, 8(4):1–55, 2007.
24. G. B. Silva. Implementação de um provador de teoremas por resolução para lógicas modais normais. Monografia de Conclusão de Curso, Bacharelado em Ciência da Computação, Universidade de Brasília, 2013. Prover available at http://www.cic.unb.br/ nalon/#software.

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.

A.1 Soundness

Soundness is given by showing that the transformation of a formula $\varphi \in \mathsf{WFF}_{\mathsf{K}_{(n)}}$ into its normal form is satisfiability preserving and that the application of the inference rules are also satisfiability preserving. Satisfiability preserving results for the transformation into SNF are given in [17, 18]. Soundness of the inference rules follows from Lemmas 1 and 2 given below.

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

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

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

Proof. Let $\mathcal{M} = (\mathcal{W}, w_0, \mathcal{R}_1, \ldots, \mathcal{R}_n, \pi)$ be a model such that (1) $\mathcal{M} \models \Box^*(C \Rightarrow \Diamond_a^{p} l')$. By $\mathbf{G}_a^{p,0,r,s}$, we have that (2) $\Diamond_a^{p} l' \Rightarrow \Box_a^{r} \Diamond_a^{s} l'$. Therefore, from (1) and (2), by the semantics of the universal operator and classical reasoning, we have that $\mathcal{M} \models \Box^*(C \Rightarrow \Box_a^{r} \Diamond_a^{s} l')$. Thus, $\mathbf{RES}_a^{p,0,r,s}$ is sound. $\Box$

Theorem 1. *The resolution-based calculi for logics of confluence are sound.*

Proof. Immediate from soundness of RES_K [18] and Lemmas 1 and 2.

A.2 Completeness

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 3. 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 [3], 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 $\Diamond_a$, $a \in \mathcal{A}_n$. Therefore, we define nodes as maximally consistent sets of literals and modal literals occurring in the set of clauses, including those literals introduced by definition clauses. That is, for any literal l occurring in the set of clauses, including definition clauses, and agents $a \in \mathcal{A}_n$, a node contains either l or $\neg l$; and either $\Box_a l$ or $\neg \Box_a l$. 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 clauses $\mathcal{T}$, where $\mathcal{N}$ is a set of nodes and each $\mathcal{E}_a$ is a set of edges labelled by $a \in \mathcal{A}_n$. 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 $\mathcal{A}_n$. An edge labelled by $a \in \mathcal{A}_n$ 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 $\eta = \eta_1$, $\eta' = \eta_k$, 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, for $k = 0$ we have that $\eta \in \mathcal{E}_a^0$, for all $\eta \in \mathcal{N}$ and $a \in \mathcal{A}_n$.

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

Definition 3. *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 $\mathcal{V} \not\models \psi$).*

A maximally consistent set of literals and modal literals contains either a propositional symbol or its negation, but not both; and it contains either a modal literal or its negation, but not both. We define satisfiability of a formula and a set of formulae with respect to a node:

Definition 4. *Let $\mathcal{V}$ be a maximal consistent set of literals and modal literals, η be a node such that η contains all the literals and modal literals in $\mathcal{V}$, φ be a Boolean combination of literals and modal 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 and modal 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. 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. 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 all literal clauses are satisfied by any node in $\mathcal{G}$. For any modal clause, $\Box^*(l' \Rightarrow m)$, where l' is a literal and m is a modal literal, delete nodes that satisfy l', but do not satisfy m. This ensures that the implications in the set $\mathcal{T}$ of clauses is satisfied. Note that the satisfaction of modal literals depend on the edges yet to be built (as given below). Also note that by satisfying the implications in the modal clauses, all definition symbols hold exactly where the corresponding modal literals hold. That is, we also delete any nodes η which do not satisfy $pos_{a,l} \Leftrightarrow \Diamond_a l$ and we have that $pos_{a,l} \in \mathcal{V}$ if and only if $\Diamond_a l \in \mathcal{V}$, for all literals l and agents $a \in \mathcal{A}_n$.

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. For each $\eta, \eta' \in \mathcal{N}$ and each $a \in \mathcal{A}_n$, there is an a-edge from η to η'. This ensures that the tautology $\mathbf{true} \Rightarrow \boxed{a}\mathbf{true}$ is satisfied by every node in $\mathcal{G}$. For every node η, delete a-edges as follows: if $\Box^*(l' \Rightarrow \boxed{a}l) \in \mathcal{T}$ and $\eta \models l'$, then delete any edges from η to η' labelled by a such that $\eta' \not\models l$. This ensures that all positive a-clauses are satisfied by any nodes in $\mathcal{G}$. Next, consider any nodes that do not satisfy the negative a-clauses in $\mathcal{T}$. For each node η and for each agent $a \in \mathcal{A}_n$, 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. This ensures that all negative a-clauses are satisfied by all nodes $\eta \in \mathcal{G}$. It also follows, by this construction, that all definition clauses are satisfied by all nodes $\eta \in \mathcal{G}$.

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.

Lemma 3. *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 the implications in modal 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 $\mathcal{M} = (\mathcal{W}, w_0, \mathcal{R}_1, \ldots, \mathcal{R}_n, \pi)$. Given the set $\mathcal{P}_\mathcal{T}$ of propositional symbols (including the definition symbols) occurring in the set of clauses $\mathcal{T}$, let $w_i \in \mathcal{W}$, where $0 \leqslant i \leqslant 2^{|\mathcal{P}_\mathcal{T}|} - 1$. There is a function $node : \mathcal{N} \to \mathcal{W}$ mapping each consistent set of literals and modal literals to names of nodes such that $node(\eta') = w_0$ for η' some initial node and each node is mapped to a different name. Let $\mathcal{R}_a = \mathcal{E}_a$ and let $\pi(w_j)(p) = true$ if, and only if, $node(\eta) = w_j$ and $p \in \eta$. $\square$

Lemma 4. *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 the 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**.

If the non-reduced graph is not empty and we have that both (1) $\Box^*(l' \Rightarrow \boxed{a} l)$ and (2) $\Box^*(l'' \Rightarrow \neg \boxed{a} l)$ are in $\mathcal{T}$, then, by construction of the graph, any node containing both l' and l'' is removed from the graph. The resolution rule **MRES** applied to (1) and (2) results in $\Box^*($**true** $\Rightarrow \neg l' \vee \neg l'')$, simulating the deletion of nodes that satisfy both l' and l''.

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 $a \in \mathcal{A}_n$, 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 $\mathbb{C}_a^\eta$ 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 $\mathbb{R}_a^\eta$ be the set of literals in the scope of $\boxed{a}$ on the right-hand side from the clauses in $\mathbb{C}_a^\eta$, that is, if $\Box^*(l_j \Rightarrow \boxed{a} l'_j) \in \mathbb{C}_a^\eta$, then $l'_j \in \mathbb{R}_a^\eta$. 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 $\neg l'$, $\mathbb{R}_a^\eta$, and the right-hand side of the literal clauses must be contradictory. As $\neg l'$ alone is not contradictory and because the case where the right-hand side of literal clauses are contradictory by themselves has been covered above (by applications of **LRES**), there are five cases:

1. Assume that $\mathbb{R}^\eta_a$ itself is contradictory. This means there must be clauses of the form $\Box^*(l_1 \Rightarrow \boxed{a} l''), \Box^*(l_2 \Rightarrow \boxed{a} \neg l'') \in \mathbb{C}^\eta_a$, where $\eta \models l_1$ and $\eta \models l_2$. Thus we can apply **NEC2** to these clauses and the negative modal clause $\Box^*(l \Rightarrow \neg \boxed{a} l')$ deriving $\Box^*(\mathbf{true} \Rightarrow \neg l_1 \vee \neg l_2 \vee \neg l)$. Hence the addition of this resolvent means that η will be deleted as required.
2. Assume that $\neg l'$ and $\mathbb{R}^\eta_a$ is contradictory. Then, $\mathbb{C}^\eta_a$ in $\mathcal{T}$ contains a clause as $\Box^*(l_1 \Rightarrow \boxed{a} l')$ where, from the definition of $\mathbb{C}^\eta_a$, $\eta \models l_1$. Thus, by an application of **MRES** to this clause and $\Box^*(l \Rightarrow \neg \boxed{a} l')$, we derive $\Box^*(\mathbf{true} \Rightarrow \neg l_1 \vee \neg l)$ and η is removed as required.
3. Assume that $\neg l'$ and the right-hand side of the literal clauses are contradictory. By applying **LRES** to the set of literal clauses, we obtain $\Box^*(\mathbf{true} \Rightarrow l')$ and use this with $\Box^*(l \Rightarrow \neg \boxed{a} l')$ to apply **NEC1** and generate $\Box^*(\mathbf{true} \Rightarrow \neg l)$ which will delete η as required.
4. Assume that $\mathbb{R}^\eta_a$ and the right-hand side of literal clauses all contribute to the contradiction (but not $\neg l'$), applying **NEC3** to the relevant clauses will delete η as required.
5. Assume that $\neg l'$, $\mathbb{R}^\eta_a$ and the right-hand side of the literal clauses all contribute to the contradiction. Thus, similarly to the above, applying **NEC1** 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 **NEC1** 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 $\boxed{a}$. The resolution rule **NEC2** deletes parts of the graph related to contradictions between the literals in the scope of $\boxed{a}$. Finally, **NEC3** deletes parts of the graph related to contradictions between the literals in the scope of $\boxed{a}$ and the set of literal clauses. These are all possible combinations of contradicting sets within 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, by Lemma 3, a model for the satisfiable set of clauses $\mathcal{T}$ can be built. $\square$

After exhaustively applying deletions to the graph, if the graph is empty, by Lemma 4, which establishes the completeness of $\mathsf{RES_K}$, there is a proof by the resolution rules shown in Section 3. For the logics of confluence, if the graph is not empty, we have to check whether we can build a model for $\mathcal{T}$, where $\mathbf{G}^{p,q,r,s}_a$ holds. The fact that this is possible is given by the following lemmas.

Lemma 5. *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 $\mathbf{RES}^{0,1,0,0}_a$.*

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 $\mathcal{T}$ as described in the completeness proof for $\mathsf{RES_K}$. We show that by applying the resolution rule $\mathbf{RES}^{0,1,0,0}_a$ any non-reflexive node in $\mathcal{G}$ is deleted. Consider a node η and $a \in \mathcal{A}$, but $(\eta, \eta) \notin \mathcal{E}_a$. There are two cases: either there are no a-edges out of η; or an a-edge leads from η but there is no a-edge from η to itself.

For the former, from the construction of the graph $\mathcal{G}$, recall that an a-edge from η to η' is only removed if $\eta \models l$, but $\eta' \not\models l'$, for some positive a-clause $\Box^*(l \Rightarrow \boxed{a} l') \in \mathcal{T}$. Therefore, if there are no a-edges out of η, 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 (1) $\Box^*(l_1 \Rightarrow \boxed{a} l'_1)$ and (2) $\Box^*(l_2 \Rightarrow \boxed{a} \neg l'_1)$ such that $\eta \models l_1$ and $\eta \models l_2$. Applying $\mathbf{RES}_a^{0,1,0,0}$ to each of these clauses we obtain (3) $\Box^*(\mathbf{true} \Rightarrow \neg l_1 \vee l'_1)$ and (4) $\Box^*(\mathbf{true} \Rightarrow \neg l_2 \vee \neg l'_1)$. By applying **LRES** to (3) and (4), we obtain (5) $\Box^*(\mathbf{true} \Rightarrow \neg l_1 \vee \neg l_2)$. Adding the resolvent (5) to the clause set deletes η from the graph. 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, where there are edges out of η but no edge from η to itself. As we have attempted to construct as many edges as possible from every node, there must be a clause as (6) $\Box^*(l_1 \Rightarrow \boxed{a} l'_1)$ such that $\eta \models l_1$ and $\eta \not\models l'_1$. By applying $\mathbf{RES}_a^{0,1,0,0}$ to (6) we obtain (7) $\Box^*(\mathbf{true} \Rightarrow \neg l'_1 \vee l_1)$. As $\eta \not\models \neg l'_1 \vee l_1$, by adding of (7) to the clause set, 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 $\Box^*(\mathbf{true} \Rightarrow D \vee l)$. $\Box$

Note that, as the conclusion of the inference rule $\mathbf{RES}_a^{0,1,0,0}$ is a propositional clause, we do not need to deal with nesting of definition symbols in this case.

We need the following lemma:

Lemma 6. *Let $\mathcal{T}$ be a satisfiable set of clauses in $\mathsf{K}_{(n)}$, l be a literal occurring in $\mathcal{T}$, where l is not a definition symbol, and G be the reduced behaviour graph for $\mathcal{T}$. Let η, η' be nodes in $\mathcal{G}$ such that $(\eta, \eta') \in \mathcal{E}_a$ and $\eta' \models l$. Then $\eta \models pos_{a,l}$.*

Proof. Assume $(\eta, \eta') \in \mathcal{E}_a$ and $\eta' \models l$, but $\eta \not\models pos_{a,l}$. As every node is a maximally consistent set of literals and modal literals, if $\eta \not\models pos_{a,l}$, then $\eta \models \neg pos_{a,l}$. By construction of $\mathcal{G}$, $\neg pos_{a,l}$ holds exactly where $\boxed{a} \neg l$ holds. Therefore, $\eta \models \boxed{a} \neg l$. By the graph construction, the edges from η to nodes that do not satisfy $\neg l$ are removed. As $\eta' \not\models \neg l$, there should be no edge $(\eta, \eta') \in \mathcal{E}_a$, which contradicts with our initial assumption. Therefore, if $(\eta, \eta') \in \mathcal{E}_a$ and $\eta' \models l$, we have that $\eta \models pos_{a,l}$. $\Box$

Lemma 7. *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 $\mathcal{T}$ as described in the completeness proof for RES_K. 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, $a \in \mathcal{A}$, $(\eta, \eta') \in \mathcal{E}_a$, 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', by Lemma 6, we have that $\eta \models pos_{a,\neg l}$. By the graph construction every node satisfies the definition clauses, therefore if $\eta \models pos_{a,\neg l}$, we also have that $\eta \models pos_{a,\neg l} \Rightarrow \neg \boxed{a} l$. 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 now that $\Box^*(l \Rightarrow \neg \boxed{a} \neg l') \in \mathcal{T}$ and $\eta \models l$. From the previous and because all modal negative modal clauses are satisfied in the graph, there must be an edge (η, η) in $\mathcal{E}_a$ and η must satisfy l', otherwise the node would have been removed. That is, 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 $\Box^*(\mathbf{true} \Rightarrow D \vee l)$ in $\mathcal{T}$. ☐

Note again that, as the conclusion of the inference rule $\mathbf{RES}_a^{1,0,0,0}$ is a propositional clause, we do not need to deal with nesting of definition symbols in this case.

Lemma 8. *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 $\mathcal{G} = \langle \mathcal{N}, \mathcal{E}_1, \ldots, \mathcal{E}_n \rangle$ for $\mathcal{T}$, as described in the completeness proof for RES_K. 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 $a \in \mathcal{A}_n$ 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 $\mathbf{RES}_a^{1,1,0,0}$ to $\Box^*(l \Rightarrow \boxed{a} l')$, 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 $\mathbf{RES}_a^{1,1,0,0}$ 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 $\eta' \not\models pos_{a,l}$, because every node is a maximally consistent set of literals and modal literals, then $\eta' \models \neg pos_{a,l}$. As every definition clause is satisfied at every node, if $\eta' \models \neg pos_{a,l}$ and $\eta' \models \neg pos_{a,l} \Rightarrow \boxed{a}\neg l$, then every edge from η' to nodes that satisfy l are removed. As $\eta \models D \vee l$ and $\eta \models \neg D$, we have that $\eta \models l$. Therefore, by construction of the graph, there is no edge (η', η) in $\mathcal{E}_a$. Now note that applying $\mathbf{RES}_a^{1,1,0,0}$ to $\Box^*(\neg pos_{a,l} \Rightarrow \boxed{a}\neg l)$ results in $\Box^*(l \Rightarrow \boxed{a} pos_{a,l})$. As $\eta \models l$, in order to satisfy $\Box^*(l \Rightarrow \boxed{a} pos_{a,l})$, the edges from η to nodes that satisfy $pos_{a,l}$ must be removed. As $\eta' \not\models pos_{a,l}$, the edge (η, η'), where $\eta \models \neg D$ and $\eta' \models \neg pos_{a,l}$ is also removed, as required. Therefore, $\mathbf{RES}_a^{0,0,1,1}$ is not needed for completeness. □

Although the resolvent of $\mathbf{RES}_a^{0,0,1,1}$ is a modal clauses, it does not refer to definition symbols. In this case, therefore, we do not need to deal with the nesting of definition symbols.

Lemma 9. *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 $\mathcal{T}$, as described in the completeness for $\mathsf{RES}_{\mathsf{K}}$. We show that by applying the inference 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. If there are no a-edges out of η, then there must be positive a-clauses of the form $\Box^*(l'_1 \Rightarrow \boxed{a} l_1)$, $\Box^*(l'_2 \Rightarrow \boxed{a} l_2)$, $\ldots$, $\Box^*(l'_k \Rightarrow \boxed{a} l_k)$, 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**, **NEC1**, **NEC2**, or **NEC3** (as described in the completeness argument for $\mathsf{K}_{(n)}$) will achieve the deletion of η. □

There is only one inference rule for serial systems and this rule does not make use of definition symbols. In this case again, we do not need to deal with nested literals.

Lemma 10. *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 $\mathbf{G}_a^{0,1,1,1}$. We construct a graph $\mathcal{G} = \langle \mathcal{N}, \mathcal{E}_1, \ldots, \mathcal{E}_n \rangle$ for $\mathcal{T}$ accordingly to the completeness argument for $\mathsf{RES}_{\mathsf{K}}$. We show that by applying the inference 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 $\mathcal{E}_a$. Note that if the right hand side of the positive modal clauses in $\mathbb{C}_a^{\eta}$ contradicted with the set of literal clauses, then there should not be an edge from η to η'. Thus, if there is no such η'', then $\mathbb{R}_a^{\eta} \cup \mathbb{R}_a^{\eta'}$ is contradictory. Therefore, there must be clauses $\Box^*(l_1 \Rightarrow \boxed{a} l)$ and $\Box^*(l_2 \Rightarrow \boxed{a} \neg l)$ such that $\eta \models l_1$ and $\eta' \models l_2$. Recall that the definition clause $\Box^*(pos_{a,l} \Rightarrow \neg \boxed{a} \neg l)$ is in the set of clauses. Applying **MRES** to $\Box^*(pos_{a,l} \Rightarrow \neg \boxed{a} \neg l)$ and $\Box^*(l_2 \Rightarrow \boxed{a} \neg l)$ results in $\Box^*(\mathbf{true} \Rightarrow \neg l_2 \vee \neg pos_{a,l})$. Applying $\mathbf{RES}_a^{0,1,1,1}$ to $\Box^*(l_1 \Rightarrow \boxed{a} l)$ results in $\Box^*(l_1 \Rightarrow \boxed{a} pos_{a,l})$. As $\Box^*(pos_{a,l_2} \Rightarrow \neg \boxed{a} \neg l_2)$ is in the set of clauses, by applying **NEC1** to $\Box^*(l_1 \Rightarrow \boxed{a} pos_{a,l})$, $\Box^*(pos_{a,l_2} \Rightarrow \neg \boxed{a} \neg l_2)$, and $\Box^*(\mathbf{true} \Rightarrow \neg l_2 \vee \neg pos_{a,l})$ we obtain $\Box^*(\mathbf{true} \Rightarrow \neg l_1 \vee \neg pos_{a,l_2})$, which is equivalent to $\Box^*(l_1 \Rightarrow \boxed{a} \neg l_2)$. Because $\eta \models l_1$ and $\eta' \not\models \neg l_2$, the edge (η, η') is removed as required.

Note that $\mathbf{RES}_a^{1,1,0,1}$ is not required for completeness. Assume that $\Box^*(l \Rightarrow \boxed{a} l')$ is in the set $\mathcal{T}$ of clauses. Applying $\mathbf{RES}_a^{1,1,0,1}$ to $\Box^*(l \Rightarrow \boxed{a} l')$ results in $\Box^*(pos_{a,l} \Rightarrow \neg \boxed{a} \neg l')$, 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 $(\eta, \eta') \in \mathcal{E}_a$ and $\eta' \models l$, by Lemma 6, we have that $\eta \models pos_{a,l}$. We consider two cases:

1. 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 $\Box^*(l \Rightarrow \boxed{a} l')$ results in $\Box^*(l \Rightarrow \boxed{a} pos_{a,l'})$. 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.
2. If there are edges $(\eta', \eta'') \in \mathcal{E}_a$, but there are no edge (η, η''), then there must be a clause as $\Box^*(l_1 \Rightarrow \boxed{a} l_2)$ 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 $\Box^*(l_1 \Rightarrow \boxed{a} l_2)$ results in $\Box^*(l_1 \Rightarrow \boxed{a} pos_{a,l_2})$. 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.

Next, we show that, although the application of $\mathbf{RES}_a^{0,1,1,1}$ to clauses whose right-hand sides are of the form $\boxed{a} pos_{a,l'}$ **(resp.** $\boxed{a} \neg pos_{a,l'}$) is correct, it is not needed for completeness. Note that applying $\mathbf{RES}_a^{0,1,1,1}$ to a clause as for instance $\Box^*(l \Rightarrow \boxed{a} pos_{a,l'})$ 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 $\Box^*(l \Rightarrow \boxed{a} pos_{a,l'})$ (resp. $\Box^*(l \Rightarrow \boxed{a} \neg pos_{a,l'})$) was obtained from previous applications of an inference rule from any of the relevant proof systems (i.e. $\mathbf{RES}_a^{1,1,0,0}$, the inference rule for **B**; $\mathbf{RES}_a^{0,1,1,1}$, the inference rule for $\mathbf{5}^{-1}$; $\mathbf{RES}_a^{1,0,1,1}$, one of the inference rules for **5**; or $\mathbf{RES}_a^{1,1,1,1}$, the inference rule for **G1**). Also assume there is a node η such that $\eta \models l$. Applying $\mathbf{RES}_a^{0,1,1,1}$ to $\Box^*(l \Rightarrow \boxed{a} pos_{a,l'})$ (resp. $\Box^*(l \Rightarrow \boxed{a} \neg pos_{a,l'})$) would result in $\Box^*(l \Rightarrow \boxed{a} pos_{a,pos_{a,l'}})$ (resp. $\Box^*(l \Rightarrow \boxed{a} pos_{a,\neg pos_{a,l'}})$). Firstly, if there are no edges out of η, then we have that adding $\Box^*(l \Rightarrow \boxed{a} pos_{a,pos_{a,l'}})$ (resp. $\Box^*(l \Rightarrow \boxed{a} pos_{a,\neg pos_{a,l'}})$) to the set of clauses will not remove further edges and, therefore, there is no need to add it in the set of clauses. Thus, assume that there is a node η', such that (η, η') in $\mathcal{E}_a$. As $\Box^*(l \Rightarrow \boxed{a} pos_{a,l'})$ (resp. $\Box^*(l \Rightarrow \boxed{a} \neg pos_{a,l'})$) is in the set of clauses and, by construction of the graph, this clause is satisfied at all nodes. Because $\eta \models l$ and $(\eta, \eta') \in \mathcal{E}_a$ we have that $\eta' \models pos_{a,l'}$ (resp. $\eta' \models \neg pos_{a,l'}$), otherwise the edge (η, η') would have been removed from the graph. By construction of the graph, 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 the edge (η, η') would have been removed from the graph. As $\Box^*(l \Rightarrow \boxed{a} pos_{a,l'})$ (resp. $\Box^*(l \Rightarrow \boxed{a} \neg pos_{a,l'})$) is in $\mathcal{T}$, $\eta \models l$, and $(\eta, \eta'') \in \mathcal{E}_a$, we have that $\eta'' \models pos_{a,l'}$ (resp. $\eta'' \models \neg pos_{a,l'}$). As $(\eta', \eta'') \in \mathcal{E}_a$, we have that $\eta'' \models \neg \boxed{a} \neg pos_{a,l'}$ (resp. $\eta'' \models \neg \boxed{a} pos_{a,l'}$), that is, $pos_{a,pos_{a,l'}}$ (resp. $pos_{a,\neg pos_{a,l'}}$) holds exactly where $pos_{a,l'}$ (resp. $\neg pos_{a,l'}$) holds. Therefore, the inclusion of $\Box^*(l \Rightarrow \boxed{a} pos_{a,pos_{a,l'}})$ (resp. $\Box^*(l \Rightarrow \boxed{a} pos_{a,\neg pos_{a,l'}})$) in the set of clauses will not cause further deletions of either edges or nodes. $\Box$

Lemma 11. *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 $\mathcal{T}$ as in the completeness proof for RES_K. We show that by applying the inference 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 $\mathcal{N}$, where $(\eta, \eta'), (\eta, \eta'') \in \mathcal{E}_a$ for some $a \in \mathcal{A}$, 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 $\eta' \models l$, by Lemma 6, 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. $\Box$

As the conclusion of $\mathbf{RES}_a^{1,0,1,0}$ does not refer to definition symbols,we do not need to deal with nested literals in this case.

Lemma 12. *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 and* $\mathbf{RES}_a^{1,0,1,1}$.

Proof. Consider any normal modal logic where each binary relation $\mathcal{R}_a$ is Euclidean. We construct a graph $\mathcal{G} = \langle \mathcal{N}, \mathcal{E}_1, \ldots, \mathcal{E}_n \rangle$ for $\mathcal{T}$, as described in the completeness proof for RES_K. We show that by applying the rule $\mathbf{RES}_a^{1,0,1,1}$ any non-Euclidean edges are deleted.

Suppose that $\Box^*(l \Rightarrow \neg \boxed{a} \neg l')$ is in the set of clauses and there is a node η such that $\eta \models l$. Then, there must be a node $\eta' \in \mathcal{N}$, such that $(\eta, \eta') \in \mathcal{E}_a$ 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', which implies that $\eta' \not\models pos_{a,l'}$. 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 $\Box^*(l \Rightarrow \boxed{a} pos_{a,l'})$. Because $\eta \models l$, we have that $\eta \models \boxed{a} pos_{a,l'}$. 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.

We show that $\mathbf{RES}_a^{1,1,1,0}$ is not needed for completeness. 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. Recall that a-edges from η' to η'' are only removed from the graph if there is a clause as $\Box^*(l \Rightarrow \boxed{a} l')$ such that $\eta' \models l$, but $\eta'' \not\models l'$. Applying **MRES** to $\Box^*(l \Rightarrow \boxed{a} l')$ and $\Box^*(pos_{a,\neg l'} \Rightarrow \neg \boxed{a} l')$, a definition clause in

$\mathcal{T}$, we obtain (1) $\Box^*(\mathbf{true} \Rightarrow \neg l \lor \neg pos_{a,\neg l'})$. Applying $\mathbf{RES}_a^{1,0,1,1}$ to $\Box^*(pos_{a,\neg l'} \Rightarrow \neg \boxed{a} l')$ results in (2) $\Box^*(pos_{a,\neg l'} \Rightarrow \boxed{a} pos_{a,\neg l'})$. Applying **NEC1** to (2), $\Box^*(pos_{a,l} \Rightarrow \neg \boxed{a} \neg l)$ (a definition clause in $\mathcal{T}$), and (1) results in $\Box^*(\mathbf{true} \Rightarrow \neg pos_{a,l} \lor \neg pos_{a,\neg l'})$. Now, as $(\eta, \eta') \in \mathcal{E}_a$ and $\eta' \models l$, by Lemma 6, $\eta \models pos_{a,l}$; also, as $(\eta, \eta'') \in \mathcal{E}_a$ and $\eta'' \models \neg l'$, by Lemma 6, $\eta \models pos_{a,\neg l}$. As $\eta \not\models \neg pos_{a,l} \lor \neg pos_{a,\neg l'}$, η is removed from the graph as required.

Next we show that, although it is correct, we do not need to apply $\mathbf{RES}_a^{1,0,1,1}$ to clauses of the form $\Box^*(l \Rightarrow \neg \boxed{a} \neg pos_{a,l'})$. Such a clause can be obtained, for instance, from an application of $\mathbf{RES}_a^{0,1,0,1}$ (the inference rule for **D**) to a clause of the form $\Box^*(l \Rightarrow \boxed{a} pos_{a,l'})$, which can be obtained from applications of other inference rules for other systems to positive modal clauses (e.g. $\mathbf{RES}_a^{1,1,1,1}$, the inference rule for **G1**). The application of $\mathbf{RES}_a^{1,0,1,1}$ to $\Box^*(l \Rightarrow \neg \boxed{a} \neg pos_{a,l'})$ would result in $\Box^*(l \Rightarrow \boxed{a} pos_{a,pos_{a,l'}})$, which removes from the graph the nodes $\eta \models l$ for each there are no η' and η'' such that both (η, η') and (η', η'') are in $\mathcal{E}_a$, but $\eta'' \not\models l'$. Assume $\Box^*(l \Rightarrow \neg \boxed{a} \neg pos_{a,l'})$ is in $\mathcal{T}$ and there is a node η such that $\eta \models l$. Because all clauses are satisfied at every node of the graph and $\eta \models l$, we have that there is a node η' such that $(\eta, \eta') \in \mathcal{E}_a$ and $\eta' \models pos_{a,l'}$. As every node satisfies the definition clauses, we have that $\eta' \models pos_{a,l'} \Rightarrow \neg \boxed{a} \neg l'$ and, by classical reasoning, $\eta' \models \neg \boxed{a} \neg l'$. Therefore, there must be a node η'' such that $(\eta', \eta'') \in \mathcal{E}_a$ and $\eta'' \models l'$. If not, η' is deleted from the graph as $\Box^*(pos_{a,l'} \Rightarrow \neg \boxed{a} \neg l')$ is not satisfied at η'; then, η is also removed from the graph as, after the removal of η', the clause $\Box^*(l \Rightarrow \neg \boxed{a} \neg pos_{a,l'})$ is not satisfied at η. Thus, the effect achieved by adding the conclusion $\Box^*(l \Rightarrow \boxed{a} pos_{a,pos_{a,l'}})$ to the clause set is the same of that of having $\Box^*(l \Rightarrow \neg \boxed{a} \neg pos_{a,l'})$ in the set of clauses. Therefore, applying $\mathbf{RES}_a^{1,0,1,1}$ to clauses of the form $\Box^*(l \Rightarrow \neg \boxed{a} \neg pos_{a,l'})$ is not needed for completeness. $\Box$

Lemma 13. *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 $\mathcal{G}$, $a \in \mathcal{A}_n$, 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 $\mathbf{RES}_a^{1,1,1,1}$.

If there is no η''' such that $(\eta', \eta''') \in \mathcal{E}_a$, then by the graph construction there must be a clause as $\Box^*(l_1 \Rightarrow \boxed{a} l_2)$ such that $\eta' \models l_1$, but $\eta''' \not\models l_2$. By applying $\mathbf{RES}_a^{1,1,1,1}$ to $\Box^*(l_1 \Rightarrow \boxed{a} l_2)$, we introduce $\Box^*(pos_{a,l_1} \Rightarrow \boxed{a} pos_{a,l_2})$ 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. Reasoning is similar, if $(\eta'', \eta''') \notin \mathcal{E}_a$.

Next, we show that we do not need to apply the inference rule $\mathbf{RES}_a^{1,1,1,1}$ to clauses of either form $\Box^*(pos_{a,l_1} \Rightarrow \boxed{a} l_2)$, $\Box^*(\neg pos_{a,l_1} \Rightarrow \boxed{a} l_2)$, $\Box^*(l_1 \Rightarrow \boxed{a} pos_{a,l_2})$, or $\Box^*(l_1 \Rightarrow \boxed{a} \neg pos_{a,l_2})$. From the above, note that the graph construction ensures that if the graph meets the frame conditions for convergent systems, then a node is either isolated (i.e. there no edges in or out of the node) or the a-edge relation is serial (taking η and η' to be the same node in the above argumentation).

First, if the node η is isolated, then by applying $\mathbf{RES}_a^{1,1,1,1}$ to the clause $\Box^*(pos_{a,l_1} \Rightarrow \boxed{a} l_2)$ (resp. $\Box^*(\neg pos_{a,l_1} \Rightarrow \boxed{a} l_2)$, $\Box^*(l_1 \Rightarrow \boxed{a} pos_{a,l_2})$, and $\Box^*(l_1 \Rightarrow \boxed{a}\neg pos_{a,l_2})$) would result in $\Box^*(pos_{a,pos_{a,l_1}} \Rightarrow \boxed{a} pos_{a,l_2})$ (resp. $\Box^*(pos_{a,\neg pos_{a,l_1}} \Rightarrow \boxed{a} pos_{a,l_2})$, $\Box^*(pos_{a,l_1} \Rightarrow \boxed{a} pos_{a,pos_{a,l_2}})$ and $\Box^*(pos_{a,l_1} \Rightarrow \boxed{a} pos_{a,\neg pos_{a,l_2}})$. Recall that positive modal clauses only remove edges from the graph. As there are no edges out of η to be further deleted, the addition of these conclusions to the clause set will not affect the a-relation with respect to isolated nodes.

Now assume there are nodes $\eta, \eta' \in \mathcal{G}$ such that $(\eta, \eta') \in \mathcal{E}_a$. By the graph construction for convergent systems, there must be a node η'' such that $(\eta', \eta'') \in \mathcal{E}_a$.

Applying $\mathbf{RES}_a^{1,1,1,1}$ to $\Box^*(pos_{a,l_1} \Rightarrow \boxed{a} l_2)$ (resp. $\Box^*(\neg pos_{a,l_1} \Rightarrow \boxed{a} l_2)$) would result in $\Box^*(pos_{a,pos_{a,l_1}} \Rightarrow \boxed{a} pos_{a,l_2})$ (resp. $\Box^*(pos_{a,\neg pos_{a,l_1}} \Rightarrow \boxed{a} pos_{a,l_2})$). The effect of adding $\Box^*(pos_{a,pos_{a,l_1}} \Rightarrow \boxed{a} pos_{a,l_2})$ (resp. $\Box^*(pos_{a,\neg pos_{a,l_1}} \Rightarrow \boxed{a} pos_{a,l_2})$) to the set of clauses is to remove from the graph any edges $(\eta, \eta') \in \mathcal{E}_a$ from a node η where $\eta \models pos_{a,pos_{a,l_1}}$ (resp. $pos_{a,\neg pos_{a,l_1}}$) to η' where $\eta' \not\models pos_{a,l_2}$. Let η be a node such that $\eta \models \neg \boxed{a} \neg pos_{a,l_1}$ (resp. $\eta \models \neg \boxed{a} pos_{a,l_1}$), which is semantically equivalent to $pos_{a,pos_{a,l_1}}$ (resp. $pos_{a,\neg pos_{a,l_1}}$). By the graph construction, there must be a node η' such that $(\eta, \eta') \in \mathcal{E}_a$ and $\eta' \models pos_{a,l_1}$ (resp. $\eta' \models \neg pos_{a,l_1}$). By the graph construction for convergent systems, as η' is not an isolated node, then there must be a node η'' such that $(\eta', \eta'') \in \mathcal{E}_a$ and $\eta'' \models l_1$. Now, because $\Box^*(pos_{a,l_1} \Rightarrow \boxed{a} l_2)$ (resp. $\Box^*(\neg pos_{a,l_1} \Rightarrow \boxed{a} l_2)$) is in the clause set, $(\eta', \eta'') \in \mathcal{E}_a$, and $\eta' \models pos_{a,l_1}$ (resp. $\eta' \models \neg pos_{a,l_1}$), we have that $\eta'' \models l_2$. By Lemma 6, $\eta' \models pos_{a,l_2}$. Therefore, by construction, if there is an a-edge from η, which satisfies $\neg \boxed{a} \neg pos_{a,l_1}$ (resp. $\eta \models \neg \boxed{a} pos_{a,l_1}$), to η'', then the existing clauses already ensure that $\eta'' \models pos_{a,l_2}$. Thus, adding $\Box^*(pos_{a,pos_{a,l_1}} \Rightarrow \boxed{a} pos_{a,l_2})$ (resp. $\Box^*(pos_{a,\neg pos_{a,l_1}} \Rightarrow \boxed{a} pos_{a,l_2})$) to the clause set will not delete further a-edges from the graph and is not needed for completeness.

Now assume that $\Box^*(l_1 \Rightarrow \boxed{a} pos_{a,l_2})$ (resp. $\Box^*(l_1 \Rightarrow \boxed{a} \neg pos_{a,l_2})$) is in the clause set. Applying $\mathbf{RES}_a^{1,1,1,1}$ to $\Box^*(l_1 \Rightarrow \boxed{a} pos_{a,l_2})$ (resp. $\Box^*(l_1 \Rightarrow \boxed{a} \neg pos_{a,l_2})$) would result in $\Box^*(pos_{a,l_1} \Rightarrow \boxed{a} pos_{a,pos_{a,l_2}})$ (resp. $\Box^*(pos_{a,l_1} \Rightarrow \boxed{a} pos_{a,\neg pos_{a,l_2}})$). Adding this to the clause set removes from the graph the a-edges from nodes η such that $\eta \models pos_{a,l_1}$ to nodes η' where $\eta' \not\models pos_{a,pos_{a,l_2}}$ (resp. $\eta' \not\models pos_{a,\neg pos_{a,l_2}}$). We show that this is not possible, i.e. if $(\eta, \eta') \in \mathcal{E}$ and $\eta \models pos_{a,l_1}$, then we have that $\eta' \models pos_{a,pos_{a,l_2}}$ (resp. $\eta' \models pos_{a,\neg pos_{a,l_2}}$). Assume there is a node η such that $\eta \models pos_{a,l_1}$. By the graph construction, there must me a node η' such that $(\eta, \eta') \in \mathcal{E}_a$ and $\eta' \models l_1$. By the graph construction for convergent systems, as η' is not an isolated node, there must be a node η'' such that $(\eta', \eta'') \in \mathcal{E}_a$. As we have that $\Box^*(l_1 \Rightarrow \boxed{a} pos_{a,l_2})$ (resp. $\Box^*(l_1 \Rightarrow \boxed{a} \neg pos_{a,l_2})$) is in the clause set and $\eta' \models l_1$, as all clauses are satisfied at all nodes, then $\eta'' \models pos_{a,l_2}$ (resp. $\eta'' \models \neg pos_{a,l_2}$). As $(\eta', \eta'') \in \mathcal{E}_a$, we have that $\eta' \models \neg \boxed{a} \neg pos_{a,l_2}$ (resp. $\eta' \models \neg \boxed{a} pos_{a,l_2}$. Therefore, no further edges will be deleted by adding $\Box^*(pos_{a,l_1} \Rightarrow \boxed{a} pos_{a,pos_{a,l_2}})$ (resp. $\Box^*(pos_{a,l_1} \Rightarrow \boxed{a} pos_{a,\neg pos_{a,l_2}})$) to the set of clauses; thus they are not needed for completeness.

Theorem 2. *Let $\mathcal{T}$ be an unsatisfiable set of clauses in $\mathbf{G}_a^{p,q,r,s}$, with $p, q, r, s \in \{0, 1\}$. A contradiction can be derived by applying the resolution rules for RES_K, given in Section 3, and Table 3.*

Proof. By Lemmas 5, 7, 8, 9, 10, 11, 12, and 13.

A.3 Termination

Termination of the proof method for logics of confluence is ensured by termination of the proof method for $\mathsf{K}_{(n)}$, given in [18], and by the fact that the resolution rules $\mathbf{RES}_a^{p,q,r,s}$ do not need to be applied to clauses which would result in nested definition symbols. Therefore, all the corresponding definition clauses can be introduced at the beginning of the proof. As a given set of clauses contains only finitely many propositional symbols, from which only finitely many SNF clauses can be constructed and therefore only finitely many new SNF clauses can be derived.

Theorem 3. *The resolution-based calculi for logics of confluence terminate.*

Proof (Sketch). From the completeness proof, the introduction of a literal such as $pos_{a,pos_{a,l}}$ for an agent a and literal l is not needed. We can show that the restrictions imposed by such clauses, together with the resolution rules for each specific logical system, are enough to ensure that the corresponding condition frame already holds. As the proof method does not introduce new literals in the clause set, there is only a finite number of clauses that can be expressed. Therefore, the proof method is terminating.