How to optimize proof-search in modal logics: a new way of proving redundancy criteria for sequent calculi

Andrei Voronkov*

Department of Computer Science
University of Manchester
voronkov@cs.man.ac.uk

Abstract

We present a bottom-up decision procedure for propositional modal logic K *based on the inverse method. The procedure is based on the "inverted" version of a sequent calculus. To restrict the search space, we prove a number of redundancy criteria for derivations in the sequent calculus. We introduce a* new technique *of proving redundancy criteria, based on the analysis of tableau-based derivations in* K. *Moreover, another new technique is used to prove completeness of proof-search with a strong notion of subsumption. This technique is based on so-called* traces. *A new formalization of the inverse method in the form of a* path calculus *considerably simplifies all proofs as compared to the previously published presentations of the inverse method. Experimental results reported elsewhere demonstrate that our method is competitive with many state-of-the-art implementations of* K.

1 Introduction

Nonclassical (propositional) logics play an increasingly important role in computer science. They are used in model checking, verification, and knowledge representation. Traditional decision procedures for these logics use semantic tableaux [Baader and Hollunder 1991, Catach 1991, Haarslev, Möller and Turhan 1998, Balsiger, Heuerding and Schwendimann 1998], optimized tableaux [Horrocks 1997, Horrocks 1998, Patel-Schneider 1998], SAT-based methods [Giunchiglia and Sebastiani 1996*a*, Giunchiglia and Sebastiani 1996*b*], or highly optimized translations into first-order classical logic [Hustadt and Schmidt 1997, Schmidt 1998].

In this paper we present a bottom-up decision procedure for propositional modal logic K based on the inverse method. All our results can be easily modified to its multimodal variant $\mathrm{K}(m)$ with m modalities, we restrict ourselves to K for simplicity. Automatic theorem proving in $\mathrm{K}(m)$ is important for several reasons, mainly because this logic is a syntactic variant of the knowledge representation formalism $\mathcal{ALC}$.

Experiments are reported in [Voronkov 2000*b*], our system K⅃ is competitive with many state-of-the-art methods for K. More precisely, on a number of benchmarks taken from the Tableau'98 competition K⅃ is behind the best two systems $\star$SAT [Tacchella 1999] and DLP [Horrocks and Patel-Schneider 1998]. On most of these benchmarks K⅃ is faster than FACT [Horrocks 1997], and K⅃ shows very different behavior from TA [Hustadt, Schmidt and Weidenbach 1998], so the two systems are very hard to compare. All other known systems were behind K⅃ on most of the benchmarks.

We believe that the technique described in our paper opens a new interesting research path in the implementation of decision procedures for other important nonclassical logics, such as temporal, description logics, or quantified boolean formulas.

This paper is organized as follows. In Sections 2, 3 and 4 we define main notions and introduce several sequent calculi for K, including an original calculus based on the notion of path in the goal formula. In Section 5 we prove several properties of derivations in the path calculus. In Section 6 we introduce the inverse path calculus and use these properties of derivations to prove that some sequents and derivations in the inverse path calculus are redundant. In Section 7 we introduce new redundancy criteria based on a strong notion of subsumption. Section 8 briefly mentions related work on the inverse method.

2 Preliminaries

In this paper we consider propositional modal logic K. K is the minimal normal modal logic, i.e. the set of modal formulas valid in all Kripke models. For the discussion of modal logics and Kripke models see any standard references [e.g. Fitting 1983, Wallen 1990]. Our technique of proving redundancies is applicable to other modal logics,

*Partially supported by grants from EPSRC and the Faculty of Science and Engineering.

at least including T, K4, S4, D and D4 and their versions with global assumptions, but the notions of redundancy for these logics are different.

We call a *literal* a propositional variable or its negation. We only consider *modal formulas in negation normal form*, i.e. formulas built from literals using $\wedge$, $\vee$, $\Box$ and $\Diamond$. A formula is called *satisfiable* if it is true in at least one Kripke model. In this paper, we will speak about (un)satisfiability of formulas instead of the dual notion of *validity*. The satisfiability problem for K is PSPACE-complete. Satisfiability in K can be characterized using a *sequent calculus* introduced below.

We denote propositional variables by lower-case letters, and formulas by upper-case letters. We will use the formula

$$E = \Box a \wedge \Box(\neg a \vee b) \wedge \Diamond\neg b \wedge \Diamond b \qquad (1)$$

for nearly all examples of this paper. We consider conjunction and disjunction right-associative, so, for example, formula (1) denotes $\Box a \wedge (\Box(\neg a \vee b) \wedge (\Diamond\neg b \wedge \Diamond b))$.

A *sequent* is a finite multiset of formulas. We denote sequents by Greek letters Γ and Δ. We denote the multiset union of two multisets Γ and Δ by Γ, Δ and the addition of an element A to a multiset Γ by Γ, A. The *multiset difference* of multisets Γ and Δ is denoted $\Gamma \dot{-} \Delta$. If an element occurs in Γ k times and in Δ l times and $k > l$, then it occurs in $\Gamma \dot{-} \Delta$ $k - l$ times, otherwise it does not occur in $\Gamma \dot{-} \Delta$ at all. If A is a member of a multiset Γ, we will write $A \dot{\in} \Gamma$. For a sequent $\Gamma = A_1, \ldots, A_n$, we denote by $\Diamond\Gamma$ the sequent $\Diamond A_1, \ldots, \Diamond A_n$. A sequent $A_1, \ldots, A_n$ is called *satisfiable* if there exists a Kripke model in which all formulas $A_1, \ldots, A_n$ are true.

A sequent Γ is a called a *subsequent* of a sequent Δ if for every formula A in Γ, if A occurs in Γ k times at the top level (i.e. as not a proper subformula of another formula), then A occurs in Δ at the top level $\geq k$ times. For example, the sequent a is a subsequent of the sequent a, a, b, but the sequent a, a is not a subsequent of the sequent a.

The *sequent calculus* Seq and an example derivation in this calculus are given in Figures 1 and 2. We call an *inference* in Seq any instance of an inference rule; *derivation of a sequent* Γ any tree formed by inferences and having Γ as the root; and *refutation of* Γ any derivation of Γ whose leaves are axioms. The term "refutation" is used because the calculus is intended for establishing unsatisfiability of formulas and sequents. We will use the notions of inference, derivation and refutation for all other calculi introduced in this paper.

It is known that Seq is complete: a sequent S is unsatisfiable if and only if it has a refutation in Seq. In addition, Seq has the *subformula property*: every formula occurring in a derivation of a sequent Γ consists of subformulas of formulas of Γ.

$$\text{axioms: } \Gamma, p, \neg p \qquad \frac{\Gamma, A, B}{\Gamma, A \wedge B}\ (\wedge)$$

$$\frac{\Gamma, A \quad \Gamma, B}{\Gamma, A \vee B}\ (\vee) \qquad \frac{\Delta, A}{\Gamma, \Box\Delta, \Diamond A}\ (\Diamond)$$

Figure 1: Sequent calculus Seq

$$\dfrac{\dfrac{\dfrac{a, \neg a, \neg b \quad a, b, \neg b}{a, \neg a \vee b, \neg b}\ (\vee)}{\Box a, \Box(\neg a \vee b), \Diamond\neg b, \Diamond b}\ (\Diamond)}{\Box a \wedge \Box(\neg a \vee b) \wedge \Diamond\neg b \wedge \Diamond b}\ (\wedge), (\wedge), (\wedge)$$

A sequence of inferences $(\wedge)$ is shown as one inference.

Figure 2: Derivation of E in Seq

3 The inverse calculus

The calculus Seq is well-known. It is straightforward to design satisfiability-checking procedures that apply the rules of Seq in the top-down way, from the goal to axioms (the tableau method). Much less known is the inverse method, based on a bottom-up version of Seq. In the inverse method, the sequent calculus depends on the goal formula G. Such a calculus, denoted by Inv^G and an example derivation in this calculus are shown in Figures 3 and 4. Compare this derivation with that of Figure 2.

The completeness of the inverse method is formulated as follows:

THEOREM 3.1 (Completeness of Inv^G) *A formula G is unsatisfiable if and only if it has a refutation in* Inv^G.

The main technique used in the proof of this theorem is the so-called *Subsequent Lemma*.

LEMMA 3.2 (Subsequent Lemma) *Let G be a formula and Γ be a sequent consisting of subformulas of G and having a refutation in* Seq. *Then there exists a subsequent Δ of Γ that has a refutation in* Inv^G. $\Box$

Completeness of Inv^G follows from this lemma immediately: indeed, if G is unsatisfiable, then the sequent G has a refutation in Seq, so by Subsequent Lemma, a subsequent of G has a refutation in Inv^G. The sequent Γ has only two subsequents: G itself, and the empty sequent, but the latter has obviously no refutation, so G has a refutation in Inv^G.

Note that the calculus Inv^G has the following properties: (i) Every formula occurring in a derivation is a subformula of G. (ii) There exists a finite number of axioms. (iii) For every sequent Γ, there exists only a finite number

$$\text{axioms: } p, \neg p \qquad \frac{\Gamma, A, A}{\Gamma, A}\ (C)$$

$$\frac{\Gamma, A}{\Gamma, A \wedge B}\ (\wedge_l) \qquad \frac{\Gamma, B}{\Gamma, A \wedge B}\ (\wedge_r)$$

$$\frac{\Gamma, A \quad \Delta, B}{\Gamma, \Delta, A \vee B}\ (\vee)$$

$$\frac{\Gamma, A}{\Box\Gamma, \Diamond A}\ (\Diamond) \qquad \frac{\Gamma}{\Box\Gamma, \Diamond A}\ (\Diamond^+)$$

In the rules, all formulas occurring in the rules are subformulas of G. The rule (C) is called the *contraction rule*.

Figure 3: Sequent calculus Inv^G

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{a, \neg a \quad b, \neg b}{a, \neg a \vee b, \neg b}\ (\vee)}{\Box a, \Box(\neg a \vee b), \Diamond\neg b}\ (\Diamond)}{\Box a,\ \ \Box(\neg a \vee b) \wedge \Diamond\neg b \wedge \Diamond b,\ \ \Box(\neg a \vee b) \wedge \Diamond\neg b \wedge \Diamond b}\ (\wedge), (\wedge), (\wedge)}{\Box a, \Box(\neg a \vee b) \wedge \Diamond\neg b \wedge \Diamond b}\ (C)}{\Box a \wedge \Box(\neg a \vee b) \wedge \Diamond\neg b \wedge \Diamond b}\ (\wedge), (\wedge), (C)$$

Some sequences of inferences are shown as one inference.

Figure 4: Derivation in Inv_E

of ways of applying $(\wedge_l)$, $(\wedge_r)$, $(\Diamond)$, $(\Diamond^+)$, or (C) to Γ. (iv) For every sequents Γ and Δ, there exists only a finite number of ways of applying $(\vee)$ to Γ and Δ. A typical proof procedure based on the inverse method uses these properties and works roughly as follows.

1. Let S be a set of sequents, initially $\{p, \neg p \mid p, \neg p \text{ are subformulas of } G\}$.
2. Repeatedly apply all possible inference rules to sequents in S, adding their conclusions to S until no new sequents can be obtained or a refutation of G is found.

However, such naive procedures are hopelessly inefficient, unless they are augmented with powerful *redundancy criteria*. There are two kinds of redundancy criteria, allowing us to get rid of *redundant sequents* and *redundant inferences*. Intuitively, a sequent is redundant if it can be removed from the search space without losing completeness. An inference is redundant if the calculus obtained by dropping this inference is still complete. Very powerful techniques for proving redundancies have been developed in resolution-based calculi for classical logic, [see e.g. Bachmair and Ganzinger 2000, Niewenhuis and Rubio 2000].

Redundancy criteria can be incompatible. For example, it is possible that some sequents Γ and Δ are redundant, but every refutation contains at least one of these sequents. Therefore, one usually proves completeness for a collection of redundancy criteria. If we prove several redundancy criteria for a calculus, the proof-search by the inverse method works as follows.

1. Let S be a set of sequents, initially $\{p, \neg p \mid p, \neg p \text{ are subformulas of } G\}$ *without the redundant sequents of this form*.
2. Repeatedly apply all *nonredundant* inferences to sequents in S, adding to S those conclusions of these inferences *that are nonredundant*.
3. Remove all sequents that *become redundant* due to the addition of these conclusions of inferences.

A number of redundancy criteria for various nonclassical logics are proved in [Voronkov 1992] using a rather general technique (though many proofs are absent). There is a paper on the inverse method for S4 [Mints, Orevkov and Tammet 1996]. This paper is interesting because it discusses many redundancy criteria, but again the proofs of many statements are missing. In particular, [Mints et al. 1996] takes it for granted that some strategies *deleting* clauses from the search space (like subsumption) are complete, though no technique for proving this has been developed.

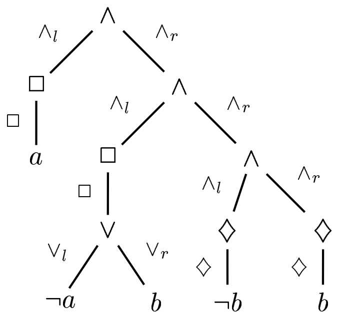
Here we present a different technique based on proving properties of refutations in Seq and transforming them into redundancy criteria for the inverse method using the Subsequent Lemma. In order to capture redundancies, we give a suitable reformulation of the sequent calculus as a *path calculus*.

4 The path calculus

For a number of reasons, we want now to speak about *occurrences* of subformulas in a goal formula G. Together with each occurrence of a subformula, we also want to keep information about Inv^G-inference rules that can be applied to the subformula corresponding to this occurrence. To achieve both aims, we introduce the notion of path in G.

A *path* is any finite sequence of symbols $\wedge_l$, $\wedge_r$, $\vee_l$, $\vee_r$, $\Box$, $\Diamond$. An example of a path is $\wedge_l\Diamond\Diamond\vee_r$. The empty path is denoted ϵ. Let G be a formula. The notion of *path in G*, or *G-path*, and the *subformula of G on a path π*, denoted $G|_\pi$, are defined inductively as follows.

- The empty path ϵ is a G-path and $G|_\epsilon$ is G.
- Suppose that π is a G-path and $G|_\pi = F$. Then
  - If F has the form $F_1 \wedge F_2$, then $\pi\wedge_l$ and $\pi\wedge_r$ are G-paths, and $G|_{\pi\wedge_l} = F_1$, $G|_{\pi\wedge_r} = F_2$. In this case we call π a $\wedge$-*path*.



π	$E\vert_\pi$
ϵ	$\Box a \wedge \Box(\neg a \vee b) \wedge \Diamond\neg b \wedge \Diamond b$
$\wedge_l$	$\Box a$
$\wedge_l \Box$	a
$\wedge_r$	$\Box(\neg a \vee b) \wedge \Diamond\neg b \wedge \Diamond b$
$\wedge_r \wedge_l$	$\Box(\neg a \vee b)$
$\wedge_r\ \wedge_l\ \Box$	$\neg a \vee b$
$\wedge_r\ \wedge_l\ \Box\vee_l$	$\neg a$
$\wedge_r\ \wedge_l\ \Box\vee_r$	b
$\wedge_r \wedge_r$	$\Diamond\neg b \wedge \Diamond b$
$\wedge_r\ \wedge_r\ \wedge_l$	$\Diamond\neg b$
$\wedge_r\ \wedge_r\ \wedge_l \Diamond$	$\neg b$
$\wedge_r\ \wedge_r\ \wedge_r$	$\Diamond b$
$\wedge_r\ \wedge_r\ \wedge_r \Diamond$	b

Figure 5: Paths and subformulas in the formula E

- If F has the form $F_1 \vee F_2$, then $\pi\vee_l$ and $\pi\vee_r$ are G-paths, and $G|_{\pi\vee_l} = F_1$, $G|_{\pi\vee_r} = F_2$. In this case we call π a $\vee$-*path*.
- If F has the form $\Box F_1$, then $\pi\Box$ is a G-path and $G|_{\pi\Box} = F_1$. In this case we call π a $\Box$-*path*.
- If F has the form $\Diamond F_1$, then $\pi\Diamond$ is a G-path and $G|_{\pi\Diamond} = F_1$. In this case we call π a $\Diamond$-*path*.

We call a path π a *literal path* if $G|_\pi$ is a literal. Note that the notions like $\wedge$-path or literal path make sense only when we speak about G-paths for a fixed formula G, and make no sense for paths in general.

The notion of subformula on a path is illustrated in Figure 5.

Evidently, for each subformula F of G, there is a G-path π such that $G|_\pi = F$. The tableau calculus Seq can be turned into a calculus where occurrences of subformulas of G are replaced by the corresponding paths. We call a *path sequent* any finite multiset of paths. When no ambiguity arises, we will refer to path sequents simply as sequents. For a path sequent $\Pi = \pi_1, \dots, \pi_n$ we denote by $\Pi\Box$ the path sequent $\pi_1\Box, \dots, \pi_n\Box$. The *path calculus* Pat^G and a derivation in it are given in Figures 6 and 7.

Completeness of Pat^G can be proved by noting bisimulation between inferences in Seq and Pat^G [see Voronkov 2000*b*].

THEOREM 4.1 (Completeness Theorem for Pat^G) *A formula G is unsatisfiable if and only if ϵ has a refutation in* Pat^G.

5 Redundancies

In order to optimize the inverse calculus, we will prove several statements about redundant sequents and derivations. To prove these statements, we will first establish several properties of derivations in the path calculus.

Let us call the *modal length* of a path the number of occurrences of $\Diamond$ and $\Box$ in the path. For example, the modal length of $\Box \wedge_l \Diamond\Diamond\vee_r$ is 3.

LEMMA 5.1 *Let Π be a path sequent occurring in a derivation of ϵ in Pat^G. Then all paths in Π have the same modal length.*

PROOF. First, observe the following fact. Let an inference in Pat^G have a conclusion Π satisfying the claim. Then all premises of this inferences satisfy the claim. Second, apply a straightforward induction on the number of inferences. $\Box$

A path π' is called a *prefix* of a path π, denoted $\pi' \sqsubseteq \pi$, if $\pi = \pi'\pi''$ for some path π''. Evidently, if $\pi' \sqsubseteq \pi$ and π is a G-path, then π' is a G-path and $G|_\pi$ is a subformula of $G|_{\pi'}$. A prefix π' of π is called *proper*, denoted $\pi' \sqsubset \pi$, if $\pi' \neq \pi$. A path sequent Π is called *prefix-free* if for every pair π_1, π_2 of different occurrences of paths in Π, π_1 is not a prefix of π_2. Note that a prefix-free sequent cannot contain two occurrences of the same path.

LEMMA 5.2 (Prefix-Freedom) *Any path sequent occurring in a derivation of ϵ is prefix-free.*

PROOF. As in the proof of Lemma 5.1, observe that this property holds for all premises of an inference, whenever it holds for its conclusion. $\Box$

We say that two paths π_1 and π_2 form a $\vee$-*fork* if one of the following conditions is satisfied:

1. $\pi_1 = \pi \vee_l \pi_1', \pi_2 = \pi \vee_r \pi_2'$ for some paths π, π_1', π_2'; or, symmetrically,
2. $\pi_1 = \pi \vee_r \pi_1', \pi_2 = \pi \vee_l \pi_2'$ for some paths π, π_1', π_2'.

The paths π_1 and π_2 are called *diamond-separated* if $\pi_1 = \pi_1'\Diamond\pi_1''$ and $\pi_2 = \pi_2'\Diamond\pi_2''$ for some paths

$$\text{axiom: } \Gamma, \pi_1, \pi_2 \qquad \frac{\Gamma, \pi\wedge_l, \pi\wedge_r}{\Gamma, \pi}\ (\wedge)$$

$$\frac{\Gamma, \pi\vee_l \quad \Gamma, \pi\vee_r}{\Gamma, \pi}\ (\vee) \qquad \frac{\Pi\Box, \pi\Diamond}{\Gamma, \Pi, \pi}\ (\Diamond)$$

All paths occurring in path sequents are G-paths. In the axioms $G|_{\pi_1} = p$ and $G|_{\pi_2} = \neg p$ for some propositional variable p.

Figure 6: Path sequent calculus Pat^G

$$\cfrac{\cfrac{\cfrac{\begin{array}{c}\wedge_l\Box,\ \wedge_r\wedge_l\Box\vee_l,\\ \wedge_r\wedge_r\wedge_l\Diamond\end{array} \qquad \begin{array}{c}\wedge_l\Box,\ \wedge_r\wedge_l\Box\vee_r,\\ \wedge_r\wedge_r\wedge_l\Diamond\end{array}}{\wedge_l\Box,\ \wedge_r\wedge_l\Box,\ \wedge_r\wedge_r\wedge_l\Diamond}\ (\vee)}{\wedge_l,\ \wedge_r\wedge_l,\ \wedge_r\wedge_r\wedge_l,\ \wedge_r\wedge_r\wedge_r}\ (\Diamond)}{\epsilon}\ (\wedge),(\wedge),(\wedge)$$

A sequence of rules $(\wedge)$ is shown as one inference.

Figure 7: Derivation in Pat^E

$\pi_1', \pi_2', \pi_1'', \pi_2''$ such that the paths π_1' and π_2' have the same modal length and $\pi_1' \neq \pi_2'$.

For example, consider the paths of Figure 5. The path $\wedge_r \wedge_l \Box\vee_l$ and $\wedge_r \wedge_l \Box\vee_r$ form a $\vee$-fork. The paths $\wedge_r \wedge_r \wedge_l \Diamond$ and $\wedge_r \wedge_r \wedge_r\Diamond$ are diamond-separated.

LEMMA 5.3 *Let* Π *be a path sequent occurring in a derivation of* ϵ *in* Pat^G. *Then no pair of paths* π_1, π_2 *in* Π *forms a* $\vee$*-fork.*

PROOF. The proof is by contradiction. We assume that two paths π_1, π_2 occurring in Π form an $\vee$-fork. Without loss of generality we assume that $\pi_1 = \pi\vee_l\pi_1'$ and $\pi_2 = \pi\vee_r\pi_2'$ for some paths π, π_1', π_2'.

Consider the branch $\mathcal{B}$ of the derivation on which Π lies and the topmost sequents on $\mathcal{B}$ where $\pi\vee_l$ and $\pi\vee_r$ first appeared. Without loss of generality assume that $\pi\vee_l$ first appeared above $\pi\vee_r$ (the other case is symmetric). Then the derivation contains the subderivation

$$\cfrac{\cfrac{\Gamma_1, \pi\vee_l \quad \Gamma_1, \pi\vee_r}{\Gamma_1, \pi}\ (\vee)}{\vdots}$$

$$\frac{\Gamma_2, \pi\vee_l \qquad \Gamma_2, \pi\vee_r}{\Gamma_2, \pi}\ (\vee)$$

Since Γ_1, π is above $\Gamma_2, \pi\vee_r$ in the derivation, Γ_2 must contain a prefix of π. This contradicts to Lemma 5.2, since the path sequent $\Gamma_2, \pi\vee_r$ should be prefix-free. $\Box$

LEMMA 5.4 *Let* Π *be a path sequent occurring in a derivation of* ϵ *in* Pat^G. *Then no pair of paths* π_1, π_2 *in* Π *is diamond-separated.*

PROOF. The proof is by contradiction. We assume that a pair of paths π_1, π_2 occurring in Π is diamond-separated. Then $\pi_1 = \pi_1'\Diamond\pi_1''$, $\pi_2 = \pi_2'\Diamond\pi_2''$, $\pi_1' \neq \pi_2'$, and paths π_1' and π_2' have the same modal length L.

Consider the branch $\mathcal{B}$ of derivation on which Π lies and the topmost sequents on $\mathcal{B}$ where $\pi_1'\Diamond$ and $\pi_2'\Diamond$ first appeared. Without loss of generality assume that $\pi_1'\Diamond$ first appeared above $\pi_2'\Diamond$ (the other case is symmetric). Then the derivation contains a subderivation

$$\cfrac{\cfrac{\Gamma_1\Box, \pi_1'\Diamond}{\Gamma_1, \pi_1', \Delta_1}\ (\Diamond)}{\vdots}$$

$$\frac{\Gamma_2\Box, \pi_2'\Diamond}{\Gamma_2, \pi_2', \Delta_2}\ (\Diamond)$$

Since π_2' has the modal length L, it is not hard to argue that all path sequents between $\Gamma_2\Box, \pi_2'\Diamond$ and $\Gamma_1, \pi_1', \Delta_1$ contain a formula with the modal length at least $L+1$. Therefore, the sequent $\Gamma_1, \pi_1', \Delta_1$ contains a path of the modal length at least $L+1$ and the path π_1' with the modal length L. This contradicts to Lemma 5.1, since all paths in a path sequent must have the same modal length. $\Box$ In view of Lemmas 5.3 and 5.4 we say that a pair of paths π_1, π_2 are *in conflict* if they either form a fork or are diamond-separated.

The lemmas above asserts some properties of all possible derivations in the path calculus. Now we want to restrict search for refutation to only a subset of derivations by introducing an ordering on the set of all G-paths.

In ordered resolution for classical logic, one can order literals in clauses and require that a resolution inference be only applied when the greatest literals in both clauses are resolved. We want to introduce a similar restriction in the modal case, by modifying classical literal orderings to their modal equivalents, called *G-orderings*. In the classical case, we can use *any* ordering on subformulas of G (or on G-paths) that respects the prefix relation. In the modal case, the situation is more complex, as we will now see: not every ordering on paths preserves completeness. For example, consider the top inference of the derivation of Figure 7:

$$\frac{\begin{array}{c}\wedge_l\Box,\ \wedge_r\wedge_l\Box\vee_l,\\ \wedge_r\wedge_r\wedge_l\Diamond\end{array} \qquad \begin{array}{c}\wedge_l\Box,\ \wedge_r\wedge_l\Box\vee_r,\\ \wedge_r\wedge_r\wedge_l\Diamond\end{array}}{\wedge_l\Box,\ \wedge_r\wedge_l\Box,\ \wedge_r\wedge_r\wedge_l\Diamond}\ (\vee)$$

It is essential in that derivation that an $(\vee)$-inference is applied above the rule $(\Diamond)$, because the rule $(\Diamond)$ is not applicable to the top sequents. However, if we impose an ordering on paths in which $\wedge_r \wedge_l \Box\vee_l$ is the smallest path in the first premise, we will rule out the possibility of applying $(\vee)$ first, and no proof will be found. So, the definition of G-ordering is more complex than in the classical case.

We call two paths *adjacent* if one of them has the form $\pi\wedge_l$ and the other $\pi\wedge_r$, or likewise, one of them has the form $\pi\vee_l$ and the other $\pi\vee_r$. For example, the following two paths in our example formula E are adjacent: $\wedge_r\ \wedge_l\ \Box\vee_l$ and $\wedge_r\ \wedge_l\ \Box\vee_r$, the formulas on this path are the two immediate subformulas of the disjunction $\neg a \vee b$. In general, every conjunction and disjunction in a formula gives rise to a pair of adjacent paths.

We denote by the generic symbol $\wedge\!\!\!\vee$ any of the symbols $\wedge$, $\vee$. Likewise, we denote by $*$ any of the symbols l, r and use notation like $\wedge_*$ or $\vee_*$. Thus $\wedge\!\!\!\vee_*$ denotes any of the symbols $\wedge_l$, $\wedge_r$, $\vee_l$, $\vee_r$. We also denote by the generic symbol $\Diamond\!\!\!\!\Box$ any of the symbols $\Diamond$, $\Box$.

Let us fix a formula G. We call a *G-ordering* any total order $\succ$ on the set of all G-paths satisfying the following conditions:

1. $\pi_1 \succ \pi_2$, whenever
   (a) the modal depth of π_1 is strictly greater than the modal depth of π_2; or
   (b) π_1, π_2 have the same modal depth, the last symbol of π_1 is $\wedge\!\!\!\vee_*$, and the last symbol of π_2 is $\Diamond\!\!\!\!\Box$; or
   (c) π_1, π_2 have the same modal depth and $\pi_2 \sqsubset \pi_1$.
2. If π_1, π_2 are adjacent, $\pi \succ \pi_1$ and π is not a prefix of π_1, then $\pi \succ \pi_2$.

It will be clear why these conditions are required of G-orderings when we prove Lemma 5.6 below. Examples of G-orderings and proof of their existence can be found in [see Voronkov 2000*b*].

We write $\pi_1 \succeq \pi_2$ to express that $\pi_1 \succ \pi_2$ or $\pi_1 = \pi_2$.

In addition to having more complex conditions on orderings for the modal case, another problem will appear in the proof of completeness. For classical logic, one can prove completeness of ordered resolution by purely semantical considerations [see e.g. Bachmair and Ganzinger 2000]. In the modal case, we want to also prove that the strategy of selecting the greatest formula (or path) in a clause is compatible with the redundancies considered so far, so we will follow the same pattern of proof: first prove a property of derivations in the tableau calculus, and then transfer it to the inverse calculus using a variant of the Subsequent Lemma. However, unlike the redundancies considered so far, such a proof of completeness will meet a technical obstacle whose essence can be expressed shortly as follows: *the ordering requirement is formulated in terms of premises of an inference, while the completeness proofs for the tableau calculus expand proofs from conclusions.*

The notion of G-ordering is introduced in order to prove the existence of refutations of a special form related to these orderings and therefore to restrict the search space to such refutations only.

For a path π and a path sequent Γ we write $\pi \succ \Gamma$ to denote $\pi \succ \pi'$ for every π' in Γ.

A $(\vee)$- (respectively, $(\wedge)$-) inference in Pat^G

$$\frac{\Gamma, \pi\vee_l \quad \Gamma, \pi\vee_r}{\Gamma, \pi}\ (\vee), \quad \text{respectively,} \quad \frac{\Gamma, \pi\wedge_l, \pi\wedge_r}{\Gamma, \pi}\ (\wedge)$$

is said to *respect a G-ordering* $\succ$ if $\pi\vee_l \succ \Gamma$ and $\pi\vee_r \succ \Gamma$ (respectively, $\pi\wedge_l \succ \Gamma$ and $\pi\wedge_r \succ \Gamma$). A derivation in Pat^G is said to respect $\succ$ if every $(\vee)$ and every $(\wedge)$-inference of this derivation respect $\succ$.

LEMMA 5.5 *Let G be a unsatisfiable formula and $\succ$ be a G-ordering. Then there exists a refutation of ϵ in Pat^G that respects $\succ$.*

It is not immediately obvious how to prove this lemma. If we try to prove this lemma by induction, starting with zero-step derivation of the empty path and trying to extend the derivation from conclusion to premises, we may be in a "deadlock" situation: i.e. obtain a path sequent Γ such that *every* $(\wedge)$- and $(\vee)$-inference having Γ as the conclusion does not respect $\succ$. The conditions on an inference to respect $\succ$ are formulated in terms of the *premises* of the inference, but in such a inductive proof we can only extend the *conclusion* by a new inference. To overcome this problem, we will try to avoid path sequents that bring us into a deadlock, by restricting ourselves to the so-called *$\succ$-compact path sequents* defined below. At the end of this section we will prove some propeerties of $\succ$-compact sequents and then prove Lemma 5.5.

Let $\pi_1, \ldots, \pi_n$ be paths in G and $\succ$ be a G-ordering. The path sequent $\Gamma = \pi_1, \ldots, \pi_n$ is called *$\succ$-compact* if for every $i = 1, \ldots, n$,

1. if π_i is a $\wedge$-path, then $\pi_i\wedge_l \succ \Gamma$ and $\pi_i\wedge_r \succ \Gamma$;
2. likewise, if π_i is a $\vee$-path, then $\pi_i\vee_l \succ \Gamma$ and $\pi_i\vee_r \succ \Gamma$.

Using our generic notation, we can reformulate the definition of $\succ$-compactness as "for every $\wedge\!\!\!\vee$-path π_i in Γ we have $\pi_i\wedge\!\!\!\vee_* \succ \Gamma$". Note that $\pi_i\wedge\!\!\!\vee_* \succ \pi_i$ is guaranteed by the conditions on G-orderings, therefore the requirement $\pi_i\wedge\!\!\!\vee_* \succ \Gamma$ can be replaced by $\pi_i\wedge\!\!\!\vee_* \succ \Gamma \setminus \{\pi_i\}$.

The next lemma asserts that a compact sequent Γ cannot lead to a deadlock. We recall that this lemma refers to the calculus Pat^G.

LEMMA 5.6 *Let $\succ$ be a G-ordering. Then (i) the premise of every $(\Diamond)$-inference is $\succ$-compact. Furthermore, let Γ be a $\succ$-compact sequent occurring in a derivation of ϵ. Then (ii) every $(\Join)$-inference having Γ as the conclusion, respects $\succ$; and (iii) if Γ contains at least one $\wedge$-path or $\vee$-path, then there exists a $(\Join)$-inference all whose premises are $\succ$-compact.*

PROOF.

1. We prove: *the premise of every $(\Diamond)$-inference*

$$\frac{\Pi\Box, \pi\Diamond}{\Gamma, \Pi, \pi}\ (\Diamond)$$

is $\succ$-compact, i.e. for every π' in the premise, if $\pi\Join_*$ is a G-path, then $\pi' \succ \Pi\Box, \pi\Diamond - \{\pi'\}$. Let us prove $\pi\Diamond\Join_* \succ \Pi\Box$, the case when $\pi' \in \Pi\Box$ is similar. By Lemma 5.1, all paths in $\Pi\Box, \pi\Diamond$ are of the same modal length, so $\pi\Diamond\Join_*$ is of the same modal length as any path in $\Pi\Box$. By condition 1b of the definition of G-ordering we have $\pi\Diamond\Join_* \succ \Pi\Box$.

2. We prove: *every $(\Join)$-inference having Γ as the conclusion, respects $\succ$.* Consider the case of $(\wedge)$-inferences, the case of $(\vee)$ is similar. Then Γ is of the form Π, π and the inference is

$$\frac{\Pi, \pi\wedge_l, \pi\wedge_r}{\Pi, \pi}\ (\wedge).$$

Since Π, π is $\succ$-compact, $\pi\wedge_l \succ \Pi, \pi$ and $\pi\wedge_r \succ \Pi, \pi$, therefore the inference respects $\succ$.

3. We assume that Γ contains at least one $\Join$-path and prove: *there exists a $(\Join)$-inference all whose premises are $\succ$-compact.* Consider the set of paths

$$S = \{\pi\Join_* \mid \pi \in \Gamma \text{ and } \pi\Join_* \text{ is a } G\text{-path}\}.$$

By the assumption, this set is nonempty. Since $\succ$ is a linear ordering, this set has the least element w.r.t. $\succ$. Without loss of generality we assume that this path is $\pi\wedge_l$. Then Γ has the form Π, π for some path sequent Π. Consider the inference

$$\frac{\Pi, \pi\wedge_l, \pi\wedge_r}{\Pi, \pi}\ (\wedge).$$

We claim that the premise of this inference is $\succ$-compact. This amounts to proving that (3a) $\pi \wedge \Join_* \succ \Pi$, whenever $\pi \wedge \Join_*$ is a G-path; and (3b) for every path $\pi' \in \Pi$ we have $\pi'\Join_* \succ \Pi, \pi\wedge$, whenever $\pi'\Join_*$ is a G-path. Let us prove this.

   (a) Since Π, π is $\succ$-compact, $\pi\wedge \succ \Pi$. Evidently, $\pi\wedge$ is a prefix of $\pi \wedge \Join_*$, thus by the definition of G-orderings (condition 1c) we have $\pi \wedge \Join_* \succ \pi\wedge$. This implies $\pi \wedge \Join_* \succ \Pi$.

   (b) Since Π, π is compact, $\pi'\Join_* \succ \Pi$, so it remains to prove $\pi'\Join_* \succ \pi\wedge$, i.e. $\pi'\Join_* \succ \pi\wedge_l$ and $\pi'\Join_* \succ \pi\wedge_r$. Note that $\pi'\Join_* \in S$ and $\pi\wedge_l$ is the least element of S, therefore $\pi'\Join_* \succeq \pi\wedge_l$. By Prefix-Freedom Lemma 5.2, $\pi'\Join_*$ and $\pi\wedge_l$ cannot coincide, thus $\pi'\Join_* \succ \pi\wedge_l$. Again by this lemma, $\pi'\Join_*$ cannot be a prefix of $\pi\wedge_l$. In addition, we have $\pi'\Join_* \succ \pi\wedge_l$ and $\pi\wedge_l$ is adjacent to $\pi\wedge_r$, then by condition 2 of the definition of G-ordering, $\pi'\Join_* \succ \pi\wedge_r$. □

This proof explains the conditions used in the definition of G-orderings. The only condition that has not been used is 1a, and indeed this condition can be omitted. This condition ensures that in the multiset ordering on sequents induced by $\succ$ the premise of any rule of Seq is greater than its conclusion.

Now we can prove Lemma 5.5: if G is unsatisfiable, then there exists a refutation of ϵ in Pat^G that respects $\succ$.

PROOF (of Lemma 5.5). The lemma is proved by induction. Starting with the path sequent ϵ, we will construct, step by step, a derivation D of ϵ such that (i) D consists only of $\succ$-compact path sequents, (ii) the formula images of all top sequents of D are unsatisfiable; until we obtain a refutation. Lemma 5.6 guarantees that the obtained refutation respects $\succ$. At every step we choose any top path sequent Π of D that is not an axiom and extend the derivation by adding an inference having Π as its conclusion.

1. (Induction base.) *The sequent ϵ is $\succ$-compact.* This means $\Join_* \succ \emptyset$, which is obvious.

2. (Induction step). Denote by Γ the formula image of Π. We have that Γ is unsatisfiable and not an axiom, and Π is $\succ$-compact. We show how to find an inference with the conclusion Π and premises satisfying the same properties.

   If Γ contains any conjunction or disjunction, then by Lemma 5.6 there exists an $(\Join)$-inference having Π as the conclusion and $\succ$-compact premises. Since $(\Join)$-rules are invertible, the formula images of these premises are also unsatisfiable.

   If Γ contains neither conjunction nor disjunction and Γ is unsatisfiable, then there is a $(\Diamond)$-inference in Seq with the conclusion Γ and unsatisfiable premise Γ'. It is not hard to argue that there exists an $(\Diamond)$-inference in Pat^G with a premise Π' such that Γ' is the formula image of Π'. By Lemma 5.6, Π' is $\succ$-compact.

$$\text{axioms: } \pi_1, \pi_2 \qquad \frac{\Gamma, \pi, \pi}{\Gamma, \pi}\ (C)$$

$$\frac{\Gamma, \pi\wedge_l}{\Gamma, \pi}\ (\wedge_l) \qquad \frac{\Gamma, \pi\wedge_r}{\Gamma, \pi}\ (\wedge_r)$$

$$\frac{\Gamma, \pi\vee_l \quad \Delta, \pi\vee_r}{\Gamma, \Delta, \pi}\ (\vee)$$

$$\frac{\Gamma\Box, \pi\Diamond}{\Gamma, \pi}\ (\Diamond) \qquad \frac{\Gamma\Box}{\Gamma, \pi}\ (\Diamond^+)$$

All paths occurring in path sequents are G-paths. In the rule (ax) we have $G|_{\pi_1} = p$ and $G|_{\pi_2} = \neg p$ for some propositional variable p. In the rule $(\Diamond^+)$, π is an $\Diamond$-path.

Figure 8: Sequent calculus IP^G

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\wedge_l\Box,\ \wedge_r\wedge_l\Box\vee_l \qquad \wedge_r\wedge_l\Box\vee_r,\ \wedge_r\wedge_r\wedge_l\Diamond}{\wedge_l\Box,\ \wedge_r\wedge_l\Box,\ \wedge_r\wedge_r\wedge_l\Diamond}\ (\vee)}{\wedge_l,\ \wedge_r\wedge_l,\ \wedge_r\wedge_r\wedge_l}\ (\Diamond)}{\wedge_l,\ \wedge_r,\ \wedge_r}\ (\wedge), (\wedge), (\wedge)}{\wedge_l,\ \wedge_r}\ (C)}{\epsilon}\ (\wedge), (\wedge), (C)$$

Some sequences of inferences are shown as one inference.

Figure 9: Derivation in IP^E

The termination of this construction of derivation can be proved by induction on the length of derivations. $\Box$

6 Inverse path calculus

In this section we define the inverse version of the path calculus, obtained by turning Inv^G into a path calculus. It is called the *inverse path calculus* IP^G and shown in Figures 8 and 9. Its completeness can also be proved by bisimulation arguments. The definition of derivations that *respect G-ordering* for IP^G is similar to that of Pat^G.

The next theorem is one of the main theorems of this paper. It asserts that a formula is unsatisfiable if and only if it has a refutation satisfying several redundancy criteria. Namely, we prove that any sequent containing a conflicting pair is redundant, and that any inference not respecting $\succ$ is redundant as well. Later, we will give an even stronger version of this statement related to the subsumption rule, but we will need to introduce a new notion of derivation first.

We denote by $\mathrm{IP}^{G,\succ}$ the calculus obtained from IP^G by (i) removal of all path sequents containing paths of different modal lengths, (ii) removal of all path sequents containing a conflicting pair of paths, (iii) removal of all inferences that do not respect $\succ$.

THEOREM 6.1 (Completeness of $\mathrm{IP}^{G,\succ}$) *A formula G is unsatisfiable if and only if ϵ has a refutation in the calculus $\mathrm{IP}^{G,\succ}$.*

The key lemma in the proof of Theorem 6.1 is as follows.

LEMMA 6.2 (Subsequent Lemma for $\mathrm{IP}^{G,\succ}$) *Let D be a refutation of ϵ in Pat^G that respects $\succ$, D'' be a subderivation in D of a sequent Γ from sequents $\Gamma_1, \ldots, \Gamma_n$, and path sequents $\Delta_1, \ldots, \Delta_n$ be subsequents of $\Gamma_1, \ldots, \Gamma_n$, respectively. Then there exists a derivation D' of Δ in $\mathrm{IP}^{G,\succ}$ from $\Delta_1, \ldots, \Delta_n$ such that Δ is a subsequent of Γ.* $\Box$

This lemma is proved in [Voronkov 2000*b*] by case analysis on the form of derivations. Using this lemma, we can prove Theorem 6.1.

PROOF (of Theorem 6.1).

($\Rightarrow$) Suppose that a formula G is unsatisfiable. By Completeness Theorem 4.1 for Pat^G, the sequent ϵ has a refutation in Pat^G. By Lemma 6.2, there exists a subsequent of ϵ that has a refutation in $\mathrm{IP}^{G,\succ}$. Evidently, the empty sequent has no refutation in IP^G, therefore ϵ has a refutation in $\mathrm{IP}^{G,\succ}$.

($\Leftarrow$) Suppose that ϵ has a refutation in $\mathrm{IP}^{G,\succ}$. Using bisimulation arguments, one can prove that G has a refutation in Inv^G and hence is unsatisfiable. $\Box$

7 More redundancies: subsumption

We say that a path sequent Π *subsumes* a path sequent Π' if Π is a subsequent of Π'. Saturation-based theorem provers for first-order logic often use subsumption as a redundancy criterion: clauses subsumed by other clauses are removed from the search space.

To prove completeness of subsumption, we need to formalize the notion of search space. In this section we give such a formalization and prove that subsumption can be used to remove clauses from the search space. The formalization here uses two main ideas: inference rules are reformulated as behaving on sets of clauses [Bachmair and Ganzinger 2000], a very general property of derivations on such sets is proved to guarantee completeness in the style of [Voronkov 1992], and then we prove that the system with subsumption satisfies these properties.

As before, we assume to deal with *inferences*, consisting of zero or more premises and one conclusion, we assume that all premises and the conclusion belong to a set of derivable objects, or simply *objects*. Examples of objects used in this paper are sequents and path sequents. An *inference rule* is a collection of inferences. An *inference system* or a *calculus* is a collection of inference rules. Alternatively, we will consider an inference system as consisting of inferences that are instances of inference rules of this system. Suppose that $\mathcal{I}$ is an inference system on objects.

We turn it into an inference system $\{\mathcal{I}\}$ on *sets of objects*, called the *set calculus* as follows. Take any inference

$$\frac{\Pi_1 \cdots \Pi_n}{\Pi} \qquad (2)$$

of $\mathcal{I}$. Then for every set S of objects the following is an inference rule of $\{\mathcal{I}\}$:

$$\frac{S \cup \{\Pi_1, \ldots, \Pi_n\}}{S \cup \{\Pi_1, \ldots, \Pi_n, \Pi\}}\ .$$

Inferences in $\{\mathcal{I}\}$ will be denoted using the $\triangleright$ sign: $S \cup \{\Pi_1, \ldots, \Pi_n\} \triangleright S \cup \{\Pi_1, \ldots, \Pi_n, \Pi\}$.

We call the *initial set of objects* of $\{\mathcal{I}\}$ the set of *all* axioms of $\mathcal{I}$. We also assume that with $\mathcal{I}$ associated is a set of objects, called the *goal objects*. For example, in all our path calculi there is one goal object: the sequent ϵ. We call a *derivation* in the set calculus any sequence of inferences:

$$S_0 \triangleright S_1 \triangleright S_2 \triangleright \ldots\ ,$$

possibly infinite, where S_0 is the initial set of objects. We say the derivation *succeeds* if some S_i contains a goal object, and *fails* otherwise.

The *subsumption strategy* that deletes a clause from the search space can now be formalized as the inference rule in the set calculus $S \cup \{\Pi_1, \Pi_2\} \triangleright S \cup \{\Pi_1\}$, provided that Π_1 subsumes Π_2.

When we have a notion of derivation that deletes objects from the search space, we can have the following problem. Suppose that an inference rule adds an object Π to the search space, and then such a deletion rule removes it. Then we may obtain an infinite failing derivation: $S \triangleright S \cup \{\Pi\} \triangleright S \triangleright S \cup \{\Pi\} \triangleright \ldots$, even when a successful derivation exists. Such a derivation is *unfair*: there have been inferences applicable to S but that were never applied. Let us formalize the notion of fairness.

Consider a derivation $S_0 \triangleright S_1 \triangleright \ldots$ in $\{\mathcal{I}\}$. The set $\bigcup_i \bigcap_{j \geq i} S_j$ is called the *limit* of this derivation. The derivation is called *fair* if, for every inference of $\mathcal{I}$ $\dfrac{\Pi_1 \cdots \Pi_n}{\Pi}$ such that $\Pi_1, \ldots, \Pi_n$ belong to the limit of the derivation, an inference of $\{\mathcal{I}\}$

$$S \cup \{\Pi_1, \ldots, \Pi_n\} \triangleright S \cup \{\Pi_1, \ldots, \Pi_n, \Pi\}$$

has been applied in the derivation, i.e., for some i and S, S_i has the form $S \cup \{\Pi_1, \ldots, \Pi_n\}$ and S_{i+1} has the form $S \cup \{\Pi_1, \ldots, \Pi_n, \Pi\}$.

It is easy to see using completeness of $\mathrm{IP}^{G,\succ}$ that $\{\mathrm{IP}^{G,\succ}\}$ is complete as well:

THEOREM 7.1 (Completeness of $\{\mathrm{IP}^{G,\succ}\}$) *Let G be a formula and $\succ$ be a G-ordering. (i) If any derivation in $\{\mathrm{IP}^{G,\succ}\}$ succeeds, then G is unsatisfiable. (ii) If G is unsatisfiable, then every fair derivation in $\{\mathrm{IP}^{G,\succ}\}$ succeeds.* $\square$

Now we are ready to prove that $\{\mathrm{IP}^{G,\succ}\}$ with subsumption is complete, i.e. every fair derivation in this system contains ϵ, whenever G is unsatisfiable. In fact, we will define an even stronger notion of subsumption first, and then prove that $\{\mathrm{IP}^{G,\succ}\}$ is complete with this stronger notion.

We say that a path sequent Π *prefix-subsumes*, a path sequent Π' if for every $\pi \mathrel{\dot\in} \Pi \mathbin{\dot-} \Pi'$ there exists a $\pi' \mathrel{\dot\in} \Pi' \mathbin{\dot-} \Pi$ such that $\pi \sqsubset \pi'$.

We call $\{\mathrm{IP}^{G,\succ}\}$ *with prefix-subsumption* the inference system obtained from $\{\mathrm{IP}^{G,\succ}\}$ by adding the *prefix-subsumption rule*:

$$S \cup \{\Pi, \Pi'\} \triangleright S \cup \{\Pi\},$$

where Π subsumes Π' and $\Pi \neq \Pi'$.

Our notion of prefix-subsumption is stronger that the standard notion of subsumption. Indeed, Π subsumes Π' if and only if $\Pi \mathbin{\dot-} \Pi'$ is empty. Therefore, if Π subsumes Π', then Π also prefix-subsumes Π'.

We will note two further properties of $\sqsubseteq$ related to proof-search optimization after proving completeness of $\{\mathrm{IP}^{G,\succ}\}$ with prefix-subsumption.

THEOREM 7.2 *(Completeness of $\{\mathrm{IP}^{G,\succ}\}$ with prefix-subsumption)*
(i) If any derivation in the system $\{\mathrm{IP}^{G,\succ}\}$ with prefix-subsumption succeeds, then G is unsatisfiable. (ii) If G is unsatisfiable, then every fair derivation $\mathcal{S}$ in $\{\mathrm{IP}^{G,\succ}\}$ with prefix-subsumption succeeds.

PROOF. (i) is proved as in Theorem 7.1 (completeness of $\{\mathrm{IP}^{G,\succ}\}$). For (ii), we will use Theorem 7.1. Let us show that (ii) immediately follows from

(3) Let $\mathcal{T}$ be a fair derivation in $\{\mathrm{IP}^{G,\succ}\}$ and a path sequent Π appears in $\mathcal{T}$. Then there exists a path sequent Γ that prefix-subsumes Π and belongs to the limit of $\mathcal{S}$.

Suppose G is unsatisfiable, then by Theorem 7.1 $\mathcal{T}$ contains the path sequent ϵ. Then by (3), $\mathcal{S}$ must contain a path sequent that prefix-subsumes ϵ, but there is only one such path sequent: ϵ itself. So, it remains to prove (3).

Suppose $\mathcal{S} = S_0 \triangleright S_1 \triangleright \ldots$ and denote by S_∞ the limit of $\mathcal{S}$. We say that a path sequent Π has been *removed* in $\mathcal{S}$ by Π' if some S_{i+1} is $S_i \setminus \{\Pi\}$, and $\Pi' \in S_{i+1}$ prefix-subsumes Π. If Π has been removed by Π_1, it is possible that Π_1 is later removed by Π_2 etc., but elementary properties of prefix-subsumption (well-foundedness) guarantee that this cannot continue infinitely often, so, starting from some j, all S_j will contain the same path sequent Π' that prefix-subsumes Π. Thus, we proved:

(4) *for every path sequent* Π *in any* S_i *there exists a* $\Pi' \in S_\infty$ *such that* Π' *prefix-subsumes* Π.

Moreover, using similar considerations one can strengthen this result to

(5) *for every path sequent* Π *in any* S_i *there exists a* $\Pi' \in S_\infty$ *such that* Π' *prefix-subsumes* Π*, and if* $\Pi'' \in S_\infty$ *prefix-subsumes* Π'*, then* $\Pi'' = \Pi'$.

For any path sequent Π (not necessarily in some S_i) we call a *trace* of Π in $\mathcal{S}$ any $\Pi' \in S_\infty$ such that Π' prefix-subsumes Π and no $\Pi'' \in S_\infty$ prefix-subsumes Π'. Now (5) can be reformulated: every $\Pi \in S_i$ has a trace in $\mathcal{S}$.

Note that (3) obviously follows from a slightly more general statement

(6) *Every path sequent* Π *that appears in a fair derivation* $\mathcal{T}$ *in* $\{\mathrm{IP}^{G,\succ}\}$ *has a trace in* $\mathcal{S}$.

We will now prove (6) and then we are done. Let $\mathcal{T} = T_0 \triangleright T_1 \triangleright \ldots$. Take the least i such that $\Pi \in T_i$. (6) will be proved by induction on i. If $i = 0$, then Π is an axiom, therefore it belongs to S_0, and by (5) has a trace in $\mathcal{S}$. If $i > 0$, then Π is obtained by an inference I of $\mathrm{IP}^{G,\succ}$ from path sequents in T_{i-1}. By the induction hypothesis, these path sequents have traces in $\mathcal{S}$. We use the case analysis on the inference I, but before we will note some general properties related to redundancies and prefix-subsumption. Namely, we prove the following 3 statements.

(7) *If every path sequent* Π *in the initial set* S_0 *contains no two paths* $\pi_1, \pi_2 \dot\in \Pi$ *with different modal lengths, then so is every path sequent* Π' *occurring in any* S_i.

(8) *Suppose* Π *is a path sequent that contains a pair of conflicting paths* π_1, π_2 *and* Π *prefix-subsumes* Π'*. Then* Π' *contains a pair of conflicting paths, too.*

(9) *Suppose* $\pi \succ \Pi'$ *and* Π *prefix-subsumes* Π'*. Then* $\pi \succ \Pi$.

(7) is straightforward. (8) follows from the following observation: if $\pi_1, \pi_2 \dot\in \Pi$ and Π prefix-subsumes Π', then there exist $\pi_1', \pi_2' \dot\in \Pi'$ such that $\pi_1 \sqsubseteq \pi_1'$ and $\pi_2 \sqsubseteq \pi_2'$, then π_1 being in conflict with π_2 implies π_1', π_2' are conflicting, too. To prove (9), note the following: Π consists of prefixes of Π', and every path is greater than any its prefix w.r.t. $\succ$, so π is greater than any path sequent in Π.

Let us now return to the proof of (6). We consider only one case (see [Voronkov 2000*b*] for other cases), when the inference I has the form

$$\frac{\Pi, \pi\wedge_l}{\Pi, \pi}\ (\wedge_l).$$

Let Γ be a trace of $\Pi, \pi\wedge_l$ in $\mathcal{S}$. Consider three possible cases:

1. *Case:* Γ *prefix-subsumes* Π. Then Γ also prefix-subsumes Π, π, and is a trace of Π, π.

2. *Case:* Γ *has the form* $\Gamma', \pi\wedge_l$*, where* Γ' *prefix-subsumes* Π. Since $\mathcal{S}$ is fair, it contains an inference

$$\frac{\Gamma', \pi\wedge_l}{\Gamma', \pi}\ (\wedge_l)$$

   By (5), $\mathcal{S}$ contains a trace of Γ', π, evidently, this trace is also a trace of Π, π. To complete the proof of this case, we have to prove that this inference is not redundant in $\mathrm{IP}^{G,\succ}$. (7) guarantees that Γ', π contains no prefixes of different modal lengths, (8) guarantees that Γ', π contains no conflicting pairs of paths, because so is Π, π, and finally, (9) guarantees that $\pi \succ \Gamma'$, because $\pi \succ \Pi$.

   Below we will not prove any more that the inferences in $\mathcal{S}$ are nonredundant, since the proof is similar to the one in this case.

3. *Case:* Γ *has the form* Γ', π'*, where* Γ' *prefix-subsumes* Π *and* $\pi' \sqsubset \pi\wedge_l$. Note that if $\pi' \sqsubset \pi$, then Γ, π' also prefix-subsumes Π, π. So we only consider the case $\pi' = \pi$. Using the definition of prefix-subsumption, one can show that either Γ', π prefix-subsumes Π, in which case we are done, or $\pi \dot\in \Gamma'$. Let us show that $\pi \dot\in \Gamma'$ is impossible. Indeed, $\pi \dot\in \Gamma'$ implies that Γ, π has a form Γ'', π, π for some Γ''. Since $\mathcal{S}$ is fair, the contraction rule should be applied in $\mathcal{S}$ to Γ'', π, π, obtaining Γ'', π which prefix-subsumes Γ'', π, π. By (5) Γ'', π has a trace in $\mathcal{S}$, then this trace prefix-subsumes Γ'', π, π. But Γ'', π, π is a trace itself, and cannot be prefix-subsumed by any other path sequent in S_∞. Contradiction. □

However, the completeness result for prefix-subsumption has some applications besides the possibility of removing subsumed sequents. Let us look at an interesting observations that come out of this result. Consider any $(\wedge_*)$-inference:

$$\frac{\Pi, \pi\wedge_*}{\Pi, \pi}\ (\wedge_*).$$

Note that the conclusion of this inference prefix-subsumes the premise. Thus, the premise can be discarded as soon as the rule is applied. Even more, when $\pi\wedge_*$ is not maximal in $\Pi, \pi\wedge_*$, so the inference is redundant, it makes sense to apply this inference, because the premise can be discarded after. If we apply $(\wedge_*)$-rules in an eager way, as soon as they are applicable, any path $\pi\wedge_*$ becomes "short-lived". This observation has a far-reaching consequence: one can modify the calculus $\mathrm{IP}^{G,\succ}$ so that such paths do not occur in sequents at all. Such an optimization first appeared for classical logic in [Voronkov 1985] published in Russian, and then in a more accessible [Voronkov 1990]. It has also been used in tableau calculi in [Degtyarev and Voronkov 1996] as a *least conjunctive superformula* optimization. It is implemented in our system KK [Voronkov 2000*a*].

8 Related works on the inverse method for nonclassical logics

Gentzen [1934] was the first to formulate the inverse method: he has shown how to prove the decidability of intuitionistic propositional logic using the bottom-up search in a sequent calculus based on the subformula property. The remark of Gentzen [1934] remained unnoticed for quite some time. The general interest to the inverse method appeared due to a series of Maslov's works, e.g. [Maslov 1971].

Voronkov [1992] sketches a general scheme of the inverse method for nonclassical logics. Some redundancy criteria considered in this paper are similar to strategies introduced in [Voronkov 1992]. However, [Voronkov 1992] contains no complete proofs.

Mints [1993] and Tammet [1994] consider the inverse method for linear logic. Mints [1994] and Tammet [1996] discuss the inverse method for intuitionistic logic, however Tammet's [1996] presentation contains several inaccuracies that are difficult to validate in the absence of proofs. For example, his general scheme of constructing inverse method calculi may result in incomplete calculi, and the intuitionistic calculus of Tammet [1996] is indeed incomplete. The ordering strategy of our paper is similar to a strategy based on the permutability of rules introduced by [Tammet 1996], with a difference that our ordering is computed in compile-time, while Tammet [1996] has to store some runtime information in derived sequents.

Mints et al. [1996] discuss the inverse method for S4 with several redundancies, but many proofs are absent, so it is difficult to compare their technique with ours.

9 Conclusion

The main contribution of this paper is that it is probably the first paper that systematically proves completeness of a calculus based on the inverse method with a large number of redundancies using a uniform technique. We note that the implementation [see Voronkov 2000*b*] based on this theory is competitive with all the best systems except *SAT and DLP, though it is not highly optimized.

1. We introduce a new technique of proving completeness of calculi with redundancies by analyzing tableau-based derivations. The previous completeness proofs were either based on ad-hoc considerations [Tammet 1996, Mints et al. 1996] or on a different technique based on a suitable notion of subsumption [Voronkov 1992][1].

2. We give a new formalization of the inverse method based on a path calculus. This formalization simplifies the definition of redundancies and completeness proofs. An example is the definition of conflicting paths and related proofs.

3. A new technique for proving completeness of subsumption-like redundancies is introduced.

A full version of this paper [Voronkov 2000*b*] discusses some relevant future research directions.

Acknowledgments. This paper benefited from numerous discussions with (in the alphabetical order) Anatoli Degtyarev, Enrico Franconi, Enrico Giunchiglia, Fausto Giunchiglia, Ian Horrocks, Fabio Massacci, Ulrike Sattler and Renate Schmidt.

References

BAADER F. AND HOLLUNDER B. [1991], A terminological knowledge representation system with complete inference algorithms, *in* H. Boley and M. Richter, eds, 'PDK'91', Vol. 567 of *Lecture Notes in Artificial Intelligence*, pp. 67–86.

BACHMAIR L. AND GANZINGER H. [2000], A theory of resolution, *in* A. Robinson and A. Voronkov, eds, 'Handbook of Automated Reasoning', Elsevier Science and MIT Press. To appear.

BALSIGER P., HEUERDING A. AND SCHWENDIMANN S. [1998], Logics Workbench 1.0, *in* H. de Swart, ed., 'Automated Reasoning with Analytic Tableaux and Related Methods, International Conference, TABLEAUX'98', Vol. 1397 of *Lecture Notes in Computer Science*, Springer Verlag, Oisterwijk, The Netherlands, p. 35.

CATACH L. [1991], 'Tableaux: a general theorem prover for modal logics', *Journal of Automated Reasoning* **7**(4), 489–510.

DEGTYAREV A. AND VORONKOV A. [1996], Equality elimination for the tableau method, *in* J. Calmet and C. Limongelli, eds, 'Design and Implementation of Symbolic Computation Systems. International Symposium, DISCO'96', Vol. 1128 of *Lecture Notes in Computer Science*, Karlsruhe, Germany, pp. 46–60.

[1]Neither [Voronkov 1992] nor [Tammet 1996] have complete proofs, which makes comparison of techniques difficult.

FITTING M. [1983], *Proof methods for modal and intuitionistic logics*, Vol. 169 of *Synthese Library*, Reidel Publ. Comp.

GENTZEN G. [1934], 'Untersuchungen über das logische Schließen', *Mathematical Zeitschrift* **39**, 176–210, 405–431. Translated as [Gentzen 1969].

GENTZEN G. [1969], Investigations into logical deduction, *in* M. Szabo, ed., 'The Collected Papers of Gerhard Gentzen', North Holland, Amsterdam, pp. 68–131. Originally appeared as [Gentzen 1934].

GIUNCHIGLIA F. AND SEBASTIANI R. [1996*a*], Building decision procedures for modal logics from propositional decision procedures: Case study of modal K, *in* M. McRobbie and J. Slaney, eds, 'CADE-13', Vol. 1104 of *Lecture Notes in Computer Science*, pp. 583–597.

GIUNCHIGLIA F. AND SEBASTIANI R. [1996*b*], A SAT-based decision procedure for $\mathcal{ALC}$, *in* 'KR'96', pp. 304–314.

HAARSLEV V., MÖLLER R. AND TURHAN A.-Y. [1998], Implementing an ALCRP(D) ABox reasoner — progress report, *in* 'Proc. DL-98 International Description Logic Workshop', Trento, Italy.

HORROCKS I. [1997], Optimizing Tableaux Decision Procedures for Modal Logic, PhD thesis, University of Manchester.

HORROCKS I. [1998], Using an expressive description logic: FaCT or fiction?, *in* A. Cohn, L. Schubert and S. Shapiro, eds, 'Principles of Knowledge Representation and Reasoning: Proceedings of the Sixth International Conference (KR'98)', Morgan Kaufmann, San Francisco, CA, pp. 636–647.

HORROCKS I. AND PATEL-SCHNEIDER P. [1998], FaCT and DLP, *in* H. de Swart, ed., 'Automated Reasoning with Analytic Tableaux and Related Methods, International Conference, TABLEAUX'98', Vol. 1397 of *Lecture Notes in Computer Science*, Springer Verlag, Oisterwijk, The Netherlands, pp. 27–30.

HUSTADT U. AND SCHMIDT R. [1997], On evaluating decision procedures for modal logic, *in* 'IJCAI-97', Vol. 1, pp. 202–207.

HUSTADT U., SCHMIDT R. AND WEIDENBACH C. [1998], Optimized functional translation and resolution, *in* H. de Swart, ed., 'TABLEAUX'98', Vol. 1397 of *Lecture Notes in Computer Science*, Oisterwijk, The Netherlands, pp. 36–37.

MASLOV S. [1971], Proof-search strategies for methods of the resolution, *in* B. Meltzer and D. Michie, eds, 'Machine Intelligence', Vol. 6, American Elsevier, pp. 77–90.

MINTS G. [1993], 'Resolution calculus for the first order linear logic', *Journal of Logic, Language and Information* **2**, 58–93.

MINTS G. [1994], Resolution strategies for the intuitionistic logic, *in* 'Constraint Programming', NATO ASI Series F, Springer Verlag, pp. 289–311.

MINTS G., OREVKOV V. AND TAMMET T. [1996], Transfer of sequent calculus strategies to resolution for S4, *in* 'Proof Theory of Modal Logic', Studies in Pure and Applied Logic, Kluwer Academic Publishers.

NIEWENHUIS R. AND RUBIO A. [2000], Paramodulation-based theorem proving, *in* A. Robinson and A. Voronkov, eds, 'Handbook of Automated Reasoning', Elsevier Science and MIT Press. To appear.

PATEL-SCHNEIDER P. [1998], DLP system description, *in* E. Franconi, G. Giacomo, R. MacGregor, W. Nutt, C. Welty and F. Sebastiani, eds, 'Collected Papers from the International Description Logic Workshop (DL'98)', Vol. 11 of *CEUR-WS*, pp. 87–89.
URL: *http://SunCITE.Informatik.RWTH-Aachen.DE/Publications/CEUR-WS*

SCHMIDT R. [1998], Resolution is a decision procedure for many propositional modal logics, *in* M. Kracht, M. de Rijke, H. Wansing and M. Zakharyaschev, eds, 'Advances in Modal Logic, I', Vol. 87 of *Lecture Notes*, CSLI Publications, Stanford, pp. 189–208.

TACCHELLA A. [1999], *sat system description, *in* P. Lambrix, A. Borgida, M. Lenzerini, R. Möller and P. Patel-Schneider, eds, 'Proceedings of the 1999 International Description Logic Workshop (DL'99)', Vol. 22 of *CEUR-WS*, pp. 142–144.
URL: *http://SunCITE.Informatik.RWTH-Aachen.DE/Publications/CEUR-WS*

TAMMET T. [1994], 'Proof strategies in linear logic', *Journal of Automated Reasoning* **12**(3), 273–304.

TAMMET T. [1996], A resolution theorem prover for intuitionistic logic, *in* M. McRobbie and J. Slaney, eds, 'CADE-13', Vol. 1104 of *Lecture Notes in Computer Science*, pp. 2–16.

VORONKOV A. [1985], A proof search method (in Russian), Vol. 107 of *Vychislitelnye Sistemy*, Novosibirsk.

VORONKOV A. [1990], A proof-search method for the first order logic, *in* P. Martin-Löf and G. Mintz, eds, 'COLOG'88', Vol. 417 of *Lecture Notes in Computer Science*, Springer Verlag, pp. 327–340.

VORONKOV A. [1992], Theorem proving in non-standard logics based on the inverse method, *in* D. Kapur, ed., '11th International Conference on Automated Deduction', Vol. 607 of *Lecture Notes in Artificial Intelligence*, pp. 648–662.

VORONKOV A. [2000*a*], How to decide K using Ǝ, *in* 'KR'2000'. To appear.

VORONKOV A. [2000*b*], How to optimize proof-search in modal logics: a new way of proving redundancy criteria for sequent calculi, Technical Report UMCS-2000-6-1, Department of Computer Science, University of Manchester.
URL: *http://www.cs.man.ac.uk/cstechrep/*

WALLEN L. [1990], *Automated Deduction in Nonclassical Logics*, The MIT Press.