Chapter 4

The Inverse Method

Anatoli Degtyarev

Andrei Voronkov

Second readers: Ullrich Hustadt, Gregory Mints, Frank Pfenning, and Renate Schmidt.

Contents

1 Introduction . . . 181
  1.1 Backward and forward reasoning . . . 181
  1.2 The inverse method . . . 181
  1.3 Structure of the chapter . . . 184
2 Preliminaries . . . 185
3 Cooking classical logic . . . 186
  3.1 Ingredients: sequents and sequent calculi . . . 186
  3.2 Subformula property of $\mathbf{C}_{inv}$ . . . 192
  3.3 The calculus $\mathbf{C}^{\xi}_{inv}$ and lifting . . . 196
  3.4 Negation normal forms . . . 203
  3.5 Saturation algorithms and linear representation of derivations . . . 206
  3.6 Exploiting other properties of sequent calculi . . . 206
4 Applying the recipe to nonclassical logics . . . 209
  4.1 Intuitionistic logic . . . 209
  4.2 Modal logics . . . 217
  4.3 Other modal logics . . . 219
5 Naming and connections with resolution . . . 219
  5.1 Naming . . . 221
  5.2 The inverse method as resolution . . . 222
  5.3 Optimized translation . . . 227
  5.4 Why hyperresolution . . . 229
  5.5 Inference rules not captured by hyperresolution . . . 229
  5.6 Further issues to naming . . . 230
  5.7 Fast food warning . . . 232
6 Season your meal: strategies and redundancies . . . 232
7 Path calculi . . . 233
  7.1 The tableau path calculus . . . 234
  7.2 Proof-theoretic properties of the tableau path calculus . . . 237
  7.3 Inverse path calculus . . . 245
  7.4 More redundancies: subsumption . . . 248
8 Logics without the contraction rules . . . 255

HANDBOOK OF AUTOMATED REASONING
Edited by Alan Robinson and Andrei Voronkov

8.1 Tableau path calculus and bisimulation lemma . . . 256
8.2 Inverse path calculus and decidability . . . 258
9 Conclusion . . . 260
9.1 History . . . 260
9.2 Limits of the technique and future research . . . 263
9.3 Warning . . . 263
Bibliography . . . 264
Index . . . 270

1. Introduction

1.1. Backward and forward reasoning

There are several methods of theorem proving. The best known methods are resolution-based and tableau-based ones. Resolution-based methods convert the goal formula into clausal norm form and perform derivation on the set of clauses using a saturation algorithm. Tableau-based methods operate directly on formulas by reducing goals to subgoals and trying to solve the subgoals, either by reduction to new subgoals or by unification-based methods.

The inverse method is much less known. Like the tableau method, it operates on formulas, but the main operation is not reduction of goals to subgoals, but rather construction of goals from previously proved subgoals. Like resolution-based methods, the inverse method uses a saturation algorithm.

1.2. The inverse method

The inverse method is unusual compared to semantic tableaux and related methods. Instead of searching for a derivation in a goal-directed manner, by reducing the goal to subgoals until all subgoals reduce to axioms, it tries to prove the goal in the inverse direction: from axioms. Since in most calculi the number of axioms is infinite, proving theorems in the inverse direction requires one to use some properties of the calculi. Similar to the tableau method, the inverse method exploits special properties of *cut-free sequent calculi*, especially the *subformula property*.

To explain why sequent calculi are most suitable for proof-search, we consider an example rule $\supset$-elimination of the natural deduction system and its analogue in a sequent calculus. The natural deduction rule has the form

$$\frac{A \quad A \supset B}{B} \ (\supset\text{-elimination}).$$

Suppose that we are given the formula B to prove. This rule says that we can try to find a formula A such that both A and $A \supset B$ are provable. There are no a priori restrictions imposed by this rule on the formula A that would allow us to restrict the search for a candidate A to a small (hopefully finite) number of formulas. Of course, one could substitute a (second-order) variable for A and try to instantiate this variable during the proof-search, like it is done in some higher-order systems, for example Isabelle [Paulson 1990], but for first-order logics this technique is not efficient compared to more specialized methods.

Consider now the corresponding rule for handling implication in the sequent calculus. We call ***signed formulas*** expressions of the form $\mathrm{T}\,A$ or $\mathrm{F}\,A$, where A is a formula. T and F are called ***signs***. Think of $\mathrm{T}\,A$ as saying that A is true, and $\mathrm{F}\,A$ as saying that A is false. A ***sequent*** is a finite multiset of signed formulas. Think of a sequent $\mathrm{T}\,A_1, \ldots, \mathrm{T}\,A_n, \mathrm{F}\,B_1, \ldots, \mathrm{F}\,B_m$ as saying that all A_i's are true and all B_j's are false. The standard sequent calculus rule for handling implication corresponding to the above natural deduction rule is

$$\frac{\Gamma,\ \mathrm{F}\,A \quad \Gamma,\ \mathrm{T}\,B}{\Gamma,\ \mathrm{T}\,A \supset B}\ (\mathrm{T} \supset),$$

where Γ is any sequent. Let us for a moment forget about Γ and consider the simplified version of this rule:

$$\frac{\mathrm{F}\,A \quad \mathrm{T}\,B}{\mathrm{T}\,A \supset B}\ (\mathrm{T} \supset).$$

Essentially, this rule says that $A \supset B$ is true in a structure if and only if A is false or B is true. Tableau-based methods would apply this rule in the backward direction: to prove the goal $\mathrm{T}\,A \supset B$, they reduce it to two subgoals $\mathrm{F}\,A$ and $\mathrm{T}\,B$. The inverse method does the inverse: if $\mathrm{F}\,A$ and $\mathrm{T}\,B$ are already established, it will establish $\mathrm{T}\,A \supset B$. Thus, the inverse method proves formulas in the *forward* direction.

At first sight this idea of the proof-search in the inverse direction reminds one of the British Museum algorithm, that generates, starting from (all possible) axioms, all provable formulas by applying all possible inference rules to already proved formulas, until the goal formula is found. Although the British Museum algorithm may in theory be better than known automated deduction methods, see [Orevkov 1983], one can hardly expect it to solve difficult problems, because of its blind, non goal-oriented behavior.

However, for some calculi we can restrict ourselves to *a strict subset* of all possible axioms and of all possible derivations. The main property of (cut-free) sequent calculi that allows us to restrict forward derivations is the ***subformula property***: if a formula has a refutation, then there exists a refutation of this formula consisting only of subformulas of this formula.

1.1. EXAMPLE. Consider the (classically valid) formula $((P \supset Q) \supset P) \supset P$, where P, Q are atomic formulas. The subformula property tells us that we can restrict ourselves to the subformulas of this formula. Since implication is the only logical connective in the formula, we should only consider sequent calculus rules for handling implication. One such rule $(\mathrm{T} \supset)$ is defined above, the other rule is

$$\frac{\Gamma,\ \mathrm{T}\,A,\ \mathrm{F}\,B}{\Gamma,\ \mathrm{F}\,A \supset B}\ (\mathrm{F} \supset).$$

Derivations in the sequent calculus search for models of sequents. This means that to establish provability of formula (1.1), they search for models of the sequent

$$\mathrm{F}\,((P \supset Q) \supset P) \supset P. \tag{1.1}$$

If this sequent has no model, then the formula $((P \supset Q) \supset P) \supset P$ is true in all structures, and therefore valid. One possible derivation of this sequent is

$$\cfrac{\cfrac{\cfrac{\mathrm{T}\,P,\ \mathrm{F}\,Q,\ \mathrm{F}\,P}{\mathrm{F}\,P \supset Q,\ \mathrm{F}\,P}\ (\mathrm{F} \supset) \qquad \mathrm{T}\,P,\ \mathrm{F}\,P}{\mathrm{T}\,(P \supset Q) \supset P,\ \mathrm{F}\,P}\ (\mathrm{T} \supset)}{\mathrm{F}\,((P \supset Q) \supset P) \supset P}\ (\mathrm{F} \supset).$$

It is quite clear how to obtain this derivation in a backward manner: starting with goal sequent (1.1), we can try to apply the inference rules (T $\supset$) and (F $\supset$) from their conclusions to their premises, until we obtain axioms.

The proof-search in the forward direction is less straightforward, but also possible. Using the subformula property of sequent calculi, we know that the only axioms that can be used in a derivation of (1.1) are sequents of the form $\Gamma, \mathrm{T}\, P, \mathrm{F}\, P$ and $\Gamma, \mathrm{T}\, Q, \mathrm{F}\, Q$. As we will see later, $\mathrm{T}\, Q$ is *not* a signed subformula of (1.1), therefore $\Gamma, \mathrm{T}\, P, \mathrm{F}\, P$ is the only choice. Since at this moment we have no idea about what Γ should be, we leave it out and start with the sequent

$$\mathrm{T}\, P,\ \mathrm{F}\, P. \tag{1.2}$$

Now, using this sequent, we can try to "build" larger signed subformulas of (1.1), using the rules of the sequent calculus but applied in the forward manner.

One appropriate signed subformula of (1.1) is $\mathrm{F}\, P \supset Q$ that can be obtained by the following inference:

$$\frac{\Gamma,\ \mathrm{T}\, P,\ \mathrm{F}\, Q}{\Gamma,\ \mathrm{F}\, P \supset Q}\ (\mathrm{F} \supset).$$

However, the only sequent proved so far is (1.2) does not contain $\mathrm{F}\, Q$. We can add $\mathrm{F}\, Q$ to (1.2) and apply this inference obtaining the sequent

$$\mathrm{F}\, P,\ \mathrm{F}\, P \supset Q. \tag{1.3}$$

This suggests that the rule $\mathrm{F} \supset$ is not well-suited for proof-search in the forward direction and can be replaced by two more convenient rules

$$\frac{\Gamma,\ \mathrm{T}\, A}{\Gamma,\ \mathrm{F}\, A \supset B}\ (\mathrm{F} \supset_l) \quad \text{and} \quad \frac{\Gamma,\ \mathrm{F}\, B}{\Gamma,\ \mathrm{F}\, A \supset B}\ (\mathrm{F} \supset_r).$$

Now we can say that (1.3) is obtained from (1.2) by the rule ($\mathrm{F} \supset_l$).

The next signed subformula of (1.1) is $\mathrm{T}\,(P \supset Q) \supset P$. We can obtain this formula using the (T $\supset$) rule of sequent calculi:

$$\frac{\Gamma,\ \mathrm{T}\, P \quad \Gamma,\ \mathrm{F}\, P \supset Q}{\Gamma,\ \mathrm{T}\,(P \supset Q) \supset P}\ (\mathrm{T} \supset).$$

We have $\mathrm{T}\, P$ in sequent (1.2) and $\mathrm{F}\, P \supset Q$ in sequent (1.3). We can apply the rule (T $\supset$) with $\mathrm{F}\, P$ substituted for Γ:

$$\frac{\mathrm{F}\, P,\ \mathrm{T}\, P \quad \mathrm{F}\, P,\ \mathrm{F}\, P \supset Q}{\mathrm{F}\, P,\ \mathrm{T}\,(P \supset Q) \supset P}\ (\mathrm{T} \supset).$$

However, it is not typical for proof-search in the forward direction that we apply the rule (T $\supset$) immediately. It is more typical that we prove sequents $\Gamma,\ \mathrm{F}\, A$ and $\Delta,\ \mathrm{T}\, B$ for *different* Γ and Δ. What we can do in such a situation is to modify the rule (T $\supset$) to be

$$\frac{\Gamma,\ \mathrm{F}\,A \quad \Delta,\ \mathrm{T}\,B}{\Gamma,\ \Delta,\ \mathrm{T}\,A \supset B}\ (\mathrm{T} \supset).$$

When we apply the modified rule to (1.2) and (1.3) we obtain

$$\mathrm{F}\,P,\ \mathrm{F}\,P,\ \mathrm{T}\,(P \supset Q) \supset P. \tag{1.4}$$

We can continue in the same manner until we obtain a derivation of the goal sequent.

This example deals with a propositional formula and is very simple. We will show later that the idea of forward proof-search in sequent calculi can be applied to the full predicate calculus using the standard three-stage ***Universal Recipe of Automated Deduction***:

- design a suitable calculus dealing with closed formulas (usually called the ground version);
- add unification and cook the calculus into a free-variable calculus;
- season it with various proof-search strategies that allow one to restrict the search space.

We will show how the Universal Recipe can be applied to various logics to obtain inverse proof procedures for these logics. Given a signed formula ξ to be proved, we define a calculus $\mathbf{C}^{\xi}_{inv}$ intended to search for a proof of ξ in the forward direction. In the sequel we always denote by ξ a closed signed formula and sometimes call it the ***goal signed formula***, or simply the ***goal***.

Convention | In the sequel ξ denotes the goal signed formula. |

The main property we want to achieve is what we call the ***finite rule property***. That is

- $\mathbf{C}^{\xi}_{inv}$ has a finite number of axioms;
- Given a finite number of sequents, there is only a finite number of inferences (i.e. one-step derivations) of $\mathbf{C}^{\xi}_{inv}$ applicable to these sequents.

Having such a calculus, a naive proof-search algorithm can, starting with the axioms of $\mathbf{C}^{\xi}_{inv}$, repeatedly apply the inference rules of the calculus to already proved sequents, until a proof of ξ is obtained or no new sequents can be derived.

1.3. Structure of the chapter

The material of this chapter is presented as a small cookbook with recipes. It is organized as follows. In Section 2 we define the main technical notions used in this chapter: those related to multisets and substitutions. In Section 3 we describe how to cook classical logic, based on the Universal Recipe. We define the ingredients for cooking: sequents and sequent calculi, the subformula property necessary for cooking, lifting, and saturation algorithms. In Section 4 we apply the Universal

Recipe to nonclassical logics: intuitionistic logic and several modal logics. In Section 5 we discuss the technique of naming and the relation of the inverse method to resolution. To season the calculi, in Section 6 we discuss the notion of redundancy in saturation-based theorem proving. In order to formalize redundancies, in Section 7 we give a new presentation of sequent calculi using the notion of paths in formulas. The use of path calculi simplifies proofs considerably, so we obtain more tasty calculi. We give proofs of completeness for several redundancies for logic **K**. In Section 8 we discuss logics without the contraction rule and show that the inverse method gives a decision procedure for first-order versions of such logics. Finally, in Section 9 we discuss the history of the inverse method, the limits of the technique, and possible future research in the area.

2. Preliminaries

Multisets. For technical reasons, we will extensively use ***finite multisets***. All precise definitions concerning multisets can be found in [Nieuwenhuis and Rubio 2001, 381] (Chapter 7 of this Handbook). If Γ is a finite multiset and γ is an element, we denote by Γ, γ or by γ, Γ the multiset obtained by adding γ to Γ. Therefore, if Γ has k copies of γ, then Γ, γ has $k+1$ copies of γ. Likewise, the union of two multisets Γ and Δ is denoted by Γ, Δ. If Γ has k copies of an element γ and Δ has l copies of γ, then Γ, Δ has $k+l$ copies of γ. Finally, we write $\Gamma \dot{\subseteq} \Delta$ to denote that Γ is a ***submultiset*** of Δ, i.e., for every element γ, if γ occurs in Γ k times, then γ occurs in Δ at least k times.

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 $A \dot{\in} \Gamma$.

Expression. By an ***expression*** we mean a term, a formula, a tuple of terms or formulas, a multiset of terms or formulas and so on. Given an expression E we denote by $var(E)$ the set of all variables occurring in E and by $free(E)$ the set of all *free* variables of E. If $var(E) = \emptyset$ then E is called ***ground***, if $free(E) = \emptyset$ then E is called ***closed***.

Substitutions. We consider a ***substitution*** as a finite mapping from variables to terms and denote it by $\{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$. It is assumed that the variables $x_1, \ldots, x_n$ are different and $x_i \neq t_i$ for all $i = 1, \ldots, n$. Given a substitution θ we denote by $dom(\theta)$ its ***domain***, i.e. the set of variables $\{x \mid \theta(x) \neq x\}$, and by $ran(\theta)$ its ***range***, i.e. the set of terms $\{\theta(x) \mid x \in dom(\theta)\}$. Thus $dom(\{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}) = \{x_1, \ldots, x_n\}$, $ran(\{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}) = \{t_1, \ldots, t_n\}$. By $vran(\theta)$ we denote the ***variable range*** of θ, i.e. the set of variables which appear in a term from $ran(\theta)$, $vran(\{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}) = var(t_1, \ldots, t_n)$. A substitution σ is ***grounding*** for a set of variables V if for every variable $v \in V$ the term $v\sigma$ is ground.

A substitution is grounding for an expression E if it is grounding for $free(E)$. The ***empty substitution***, denoted ε, is the only substitution with $dom(\varepsilon) = \emptyset$.

A substitution $\theta = \{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$ is ***admissible*** for an expression E, if E has no occurrences of a formula $\forall y A$ or $\exists y A$ such that for some i we have $x_i \in free(\forall y A)$ or $x_i \in free(\exists y A)$, and $y \in var(t_i)$. In other words, for every $i \in \{1, \ldots, n\}$ the ***term*** t_i ***is free for*** x_i ***in*** E, see [Kleene 1967, page 94].

For an expression E and a substitution θ admissible for E, $E\theta$ stands for the result of applying θ to E. It is obtained from E by the simultaneous replacement of free occurrences of $x_1, \ldots, x_n$ by $t_1, \ldots, t_n$, respectively.

Let E be an expression. Following [Kleene 1967], whenever we introduce a notation like $E(x_1, \ldots, x_n)$, $n \geq 1$, showing variables $x_1, \ldots, x_n$ in the notation, we will thereafter understand by $E(t_1, \ldots, t_n)$ the result of application of the substitution $\theta = \{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$ to E under the condition that θ is admissible for E. Of course, it is not required that the expression denoted by $E(x_1, \ldots, x_n)$ actually contains x_i free, also it is not excluded that $E(x_1, \ldots, x_n)$ may contain free variables other than $x_1, \ldots, x_n$.

The notation $\rightleftharpoons$ stands for "equal by definition".

The ***composition of substitutions*** σ and θ, denoted $\sigma\theta$, is the substitution defined by $x(\sigma\theta) \rightleftharpoons (x\sigma)\theta$, for every variable x.

Let σ_1, σ_2 be substitutions such that $dom(\sigma_1) \cap dom(\sigma_2) = \emptyset$. By $\sigma_1 \cup \sigma_2$ we denote the substitution σ such that $dom(\sigma) \rightleftharpoons dom(\sigma_1) \cup dom(\sigma_2)$ and

$$x\sigma \rightleftharpoons \begin{cases} x\sigma_1, & \text{if } x \in dom(\sigma_1), \\ x\sigma_2, & \text{if } x \in dom(\sigma_2), \end{cases}$$

for all $x \in dom(\sigma)$.

We will need more definitions concerning substitutions and unification. They will be introduced later, as soon as they are needed.

3. Cooking classical logic

In this section we apply the universal recipe of automated deduction to cook the inverse method for classical logic.

3.1. Ingredients: sequents and sequent calculi

3.1. DEFINITION *(truth, satisfiability, validity).* A closed signed formula $\mathrm{T}\,A$ (respectively, $\mathrm{F}\,A$) is ***true*** in a structure if so is the formula A (respectively, $\neg A$). A closed sequent is ***true*** in a structure if so is every signed formula in this sequent. If a closed signed formula (respectively, a closed sequent) is true in a structure, we say that the structure is a ***model*** of the signed formula (respectively, the sequent). A sequent is ***satisfiable*** if it has a model.

$$\frac{}{\mathrm{T}\,A, \mathrm{F}\,A}\ (Ax)$$

$$\frac{\Gamma,\ \gamma,\ \gamma}{\Gamma,\ \gamma}\ (C)$$

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

$$\frac{\Gamma,\ \mathrm{F}\,A \quad \Delta,\ \mathrm{F}\,B}{\Gamma,\ \Delta,\ \mathrm{F}\,A \wedge B}\ (\mathrm{F} \wedge)$$

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

$$\frac{\Gamma,\ \mathrm{F}\,A}{\Gamma,\ \mathrm{F}\,A \vee B}\ (\mathrm{F} \vee_l) \qquad \frac{\Gamma,\ \mathrm{F}\,B}{\Gamma,\ \mathrm{F}\,A \vee B}\ (\mathrm{F} \vee_r)$$

$$\frac{\Gamma,\ \mathrm{F}\,A \quad \Delta,\ \mathrm{T}\,B}{\Gamma,\ \Delta,\ \mathrm{T}\,A \supset B}\ (\mathrm{T} \supset)$$

$$\frac{\Gamma,\ \mathrm{T}\,A}{\Gamma,\ \mathrm{F}\,A \supset B}\ (\mathrm{F} \supset_l) \qquad \frac{\Gamma,\ \mathrm{F}\,B}{\Gamma,\ \mathrm{F}\,A \supset B}\ (\mathrm{F} \supset_r)$$

$$\frac{\Gamma,\ \mathrm{F}\,A}{\Gamma,\ \mathrm{T}\,\neg A}\ (\mathrm{T} \neg) \qquad \frac{\Gamma,\ \mathrm{T}\,A}{\Gamma,\ \mathrm{F}\,\neg A}\ (\mathrm{F} \neg)$$

$$\frac{\Gamma,\ \mathrm{T}\,A(t)}{\Gamma,\ \mathrm{T}\,\forall x A(x)}\ (\mathrm{T} \forall) \qquad \frac{\Gamma,\ \mathrm{F}\,A(y)}{\Gamma,\ \mathrm{F}\,\forall x A(x)}\ (\mathrm{F} \forall)$$

$$\frac{\Gamma,\ \mathrm{T}\,A(y)}{\Gamma,\ \mathrm{T}\,\exists x A(x)}\ (\mathrm{T} \exists) \qquad \frac{\Gamma,\ \mathrm{F}\,A(t)}{\Gamma,\ \mathrm{F}\,\exists x A(x)}\ (\mathrm{F} \exists)$$

In the rule (Ax), A is an atomic formula. In the rules $(\mathrm{F}\,\forall)$ and $(\mathrm{T}\,\exists)$ the variable y is free for x in $A(x)$ and $A(y) = A(x)\{x \mapsto y\}$, furthermore y has no free occurrences in the conclusions of the rules. The variable y is called the ***eigenvariable*** of these rules. This condition on the rules $(\mathrm{F}\,\forall)$ and $(\mathrm{T}\,\exists)$ is called the ***eigenvariable condition***. The rule (C) is called ***contraction***.

Figure 1: Calculus $\mathbf{C}_{inv}$

All sequent calculi introduced in this chapter establish the unsatisfiability of sequents. In view of this fact, proofs in these calculi will sometimes be called *refutations*.

We will denote sequents by Γ, Δ, ∇. A detailed discussion of sequent calculi can be found in [Degtyarev and Voronkov 2001*a*] (Chapter 10 of this Handbook). The *sequent calculus* $\mathbf{C}_{inv}$ for classical logic is shown in Figure 1.

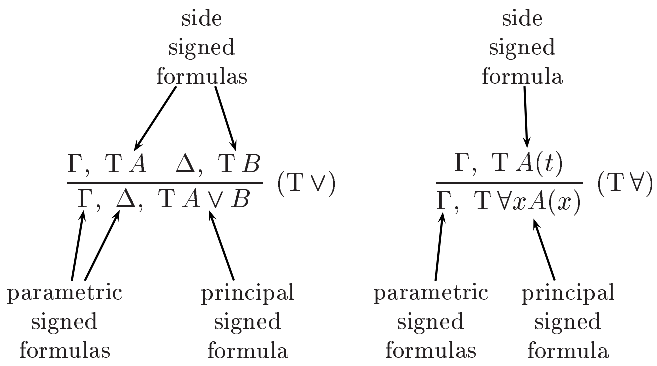


Figure 2: Principal, side and parametric signed formulas

3.2. Note. The calculus $\mathbf{C}_{inv}$ is one of the calculi developed from the original calculus LK of Gentzen [1934]. In formalizations of sequent calculi developed for semantic tableaux, LK is usually defined using *invertible rules*, following calculi introduced in [Kanger 1957, Kanger 1963], see [Degtyarev and Voronkov 2001*a*] (Chapter 10 of this Handbook). The discussion in the introduction on pages 182–184 reveals why we formulate the rules of $\mathbf{C}$ in a noninvertible way.

Let us introduce some standard definitions related to rules of the sequent calculi illustrated by Figure 2, adapted from [Kleene 1967]. Every rule infers a sequent from one or two sequents.

3.3. Definition *(principal, side, and parametric formulas).* The new signed formula introduced by the rule is called the ***principal signed formula*** of the rule. The signed formula or formulas that occur in the premises of the rule and do not occur in the conclusion are called the ***side signed formulas*** of this rule. All other signed formulas, denoted by Γ, Δ, are called ***parametric signed formulas*** of the rule.

Since we will establish a lot of proof-theoretic results about derivations in various sequent calculi, we introduce terminology related to derivations. A similar terminology is used in [Bachmair and Ganzinger 2001] (Chapter 2 of this Handbook).

3.4. Definition *(inference, derivation, refutation).* An ***inference*** in $\mathbf{C}_{inv}$ is any instance of an inference rule. A ***derivation of a sequent*** Γ is any tree of sequents formed by inferences and having Γ as the root. A ***refutation of*** Γ is any derivation of Γ whose leaves are axioms.

We will use the notions of inference, derivation and refutation for all other calculi introduced in this chapter. When we give examples of derivations, we will put side formulas of all inferences in shaded boxes. We will also omit the inferences by (Ax).

3.5. EXAMPLE. Consider the refutation of formula (1.1) in $\mathbf{C}_{inv}$:

$$
\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\mathrm{T}\,P,\ \mathrm{F}\,P}{\mathrm{F}\,P \supset Q,\ \mathrm{F}\,P}\,(\mathrm{F}\supset_l) \qquad \mathrm{T}\,P,\ \mathrm{F}\,P}{\mathrm{T}\,(P \supset Q) \supset P,\ \mathrm{F}\,P,\ \mathrm{F}\,P}\,(\mathrm{T}\supset)}{\mathrm{T}\,(P \supset Q) \supset P,\ \mathrm{F}\,P}\,(C)}{\mathrm{F}\,((P \supset Q) \supset P) \supset P,\ \mathrm{F}\,P}\,(\mathrm{F}\supset_l)}{\mathrm{F}\,((P \supset Q) \supset P) \supset P,\ \mathrm{F}\,((P \supset Q) \supset P) \supset P}\,(\mathrm{F}\supset_r)}{\mathrm{F}\,((P \supset Q) \supset P) \supset P}\,(C)
$$

Suppose we have a calculus C for some logic L. Logics are usually defined using a suitable semantics, in this chapter we assume that the semantics defines the notion of ***satisfiable sequent***. A calculus C is called ***sound*** for L is every sequent that has a refutation in C is unsatisfiable in L. A calculus C is called ***complete for*** L if every unsatisfiable sequent of L has a refutation in C. Sometimes, we will deal with weaker notions of completeness, when only unsatisfiable sequents consisting of one formula are guaranteed to have a refutation. Usually, inference rules in a sound calculus have the following property (which guarantees soundness): if all premises of this rule are unsatisfiable, then so is the conclusion. Since soundness will be more or less obvious for most calculi of this chapter, we will mainly speak about completeness, even when we mean soundness and completeness.

An inference rule is called ***invertible*** if it has the converse property: if the conclusion of the rule is unsatisfiable, then so are all premises of the rule. Many standard completeness rules for tableau calculi are based on calculi in which all or nearly all inference rules are invertible.

Since the rules of $\mathbf{C}_{inv}$ are not invertible, the standard completeness proofs are not easily applicable to it. Moreover, $\mathbf{C}_{inv}$ is *incomplete* for refuting sequents, but complete for refuting formulas. Let us show the technique that allows to establish completeness by purely proof-theoretic arguments. The technique is based on the following observation.

Take any cut-free sequent calculus for classical logic that is known to be complete. As an example, we take the calculus $\mathbf{C}_{tab}$ with invertible rules used in some formalizations of the tableau method. The *calculus* $\mathbf{C}_{tab}$ is shown in Figure 3. This calculus is essentially the calculus of Kanger [1957] or $G4$ of Kleene [1967]. The invertibility of rules of $\mathbf{C}_{tab}$ allows us to prove completeness using the standard arguments based on the Hintikka sets [Hintikka 1955], see also [Kanger 1957, Kleene 1967, Smullyan 1968, Fitting 1996].

3.6. THEOREM (Completeness of $\mathbf{C}_{tab}$). *Let Γ be a closed sequent. Γ is unsatisfiable if and only if there exists a refutation of Γ in $\mathbf{C}_{tab}$.* □

The rules of $\mathbf{C}_{tab}$, except for the rules for negation and two of the quantifier rules, are different from those of $\mathbf{C}_{inv}$. Let us think how we can use completeness of $\mathbf{C}_{tab}$

$$\frac{}{\Gamma,\ \mathrm{T}\,A,\ \mathrm{F}\,A}\ (Ax)$$

$$\frac{\Gamma,\ \mathrm{T}\,A,\ \mathrm{T}\,B}{\Gamma,\ \mathrm{T}\,A\wedge B}\ (\mathrm{T}\wedge) \qquad \frac{\Gamma,\ \mathrm{F}\,A\quad \Gamma,\ \mathrm{F}\,B}{\Gamma,\ \mathrm{F}\,A\wedge B}\ (\mathrm{F}\wedge)$$

$$\frac{\Gamma,\ \mathrm{T}\,A\quad \Gamma,\ \mathrm{T}\,B}{\Gamma,\ \mathrm{T}\,A\vee B}\ (\mathrm{T}\vee) \qquad \frac{\Gamma,\ \mathrm{F}\,A,\ \mathrm{F}\,B}{\Gamma,\ \mathrm{F}\,A\vee B}\ (\mathrm{F}\vee)$$

$$\frac{\Gamma,\ \mathrm{F}\,A\quad \Gamma,\ \mathrm{T}\,B}{\Gamma,\ \mathrm{T}\,A\supset B}\ (\mathrm{T}\supset) \qquad \frac{\Gamma,\ \mathrm{T}\,A,\ \mathrm{F}\,B}{\Gamma,\ \mathrm{F}\,A\supset B}\ (\mathrm{F}\supset)$$

$$\frac{\Gamma,\ \mathrm{F}\,A}{\Gamma,\ \mathrm{T}\,\neg A}\ (\mathrm{T}\,\neg) \qquad \frac{\Gamma,\ \mathrm{T}\,A}{\Gamma,\ \mathrm{F}\,\neg A}\ (\mathrm{F}\,\neg)$$

$$\frac{\Gamma,\ \mathrm{T}\,\forall x A(x),\ \mathrm{T}\,A(t)}{\Gamma,\ \mathrm{T}\,\forall x A(x)}\ (\mathrm{T}\,\forall) \qquad \frac{\Gamma,\ \mathrm{F}\,A(y)}{\Gamma,\ \mathrm{F}\,\forall x A(x)}\ (\mathrm{F}\,\forall)$$

$$\frac{\Gamma,\ \mathrm{T}\,A(y)}{\Gamma,\ \mathrm{T}\,\exists x A(x)}\ (\mathrm{T}\,\exists) \qquad \frac{\Gamma,\ \mathrm{F}\,\exists x A(x),\ \mathrm{F}\,A(t)}{\Gamma,\ \mathrm{F}\,\exists x A(x)}\ (\mathrm{F}\,\exists)$$

In the rule (Ax), A is an atomic formula. The rules $(\mathrm{F}\,\forall)$ and $(\mathrm{T}\,\exists)$ satisfy the eigenvariable condition.

Figure 3: Calculus $\mathbf{C}_{tab}$

to prove completeness of $\mathbf{C}_{inv}$. We have a problem already with axioms. Axioms of $\mathbf{C}_{tab}$ have the form

$$\frac{}{\Gamma,\ \mathrm{T}\,A,\ \mathrm{F}\,A}\ (Ax).$$

As a matter of fact, *not every* sequent $\Gamma,\ \mathrm{T}\,A,\ \mathrm{F}\,A$ has a refutation in $\mathbf{C}_{inv}$. A simplest example is $\mathrm{T}\,P,\ \mathrm{F}\,P,\ \mathrm{T}\,Q$, where P, Q are distinct 0-ary relation symbols. (Note that this shows that not every unsatisfiable sequent has a refutation in $\mathbf{C}_{inv}$.) However, we can derive all sequents of the form $\mathrm{T}\,A,\ \mathrm{F}\,A$ using the axioms of $\mathbf{C}_{inv}$. This means that for every axiom Δ of $\mathbf{C}_{tab}$ there exists $\Delta' \dot{\subseteq} \Delta$ such that Δ' has a refutation in $\mathbf{C}_{inv}$. We generalize this observation to the following fact that will imply completeness of $\mathbf{C}_{inv}$ for formulas in a rather straightforward way.

3.7. LEMMA (Subsequent Lemma). *For every sequent Γ that has a refutation in $\mathbf{C}_{tab}$ there exists a sequent Δ such that $\Delta \dot{\subseteq} \Gamma$ and Δ has refutation in $\mathbf{C}_{inv}$.*

PROOF. The proof is by induction on the length of a refutation of Γ in $\mathbf{C}_{tab}$. We consider several cases, the other cases will be similar.

CASE (Ax). This case is obvious.

CASE $(\mathrm{F}\supset)$. The rule of $\mathbf{C}_{tab}$ has the form

$$\frac{\Gamma,\ \mathrm{T}\,A,\ \mathrm{F}\,B}{\Gamma,\ \mathrm{F}\,A \supset B}\ (\mathrm{F} \supset).$$

By the induction hypothesis, there exists a sequent $\Delta \mathrel{\dot{\subseteq}} (\Gamma,\ \mathrm{T}\,A,\ \mathrm{F}\,B)$ which has a refutation in $\mathbf{C}_{inv}$. We have to find a sequent $\Delta' \mathrel{\dot{\subseteq}} (\Gamma,\ \mathrm{F}\,A \supset B)$ which has a refutation in $\mathbf{C}_{inv}$. Consider the following cases, depending on whether $\mathrm{T}\,A$ and $\mathrm{F}\,B$ occur in Δ.

SUBCASE $((\mathrm{T}\,A, \mathrm{F}\,B) \mathrel{\dot{\subseteq}} \Delta)$. We have $\Delta = (\nabla,\ \mathrm{T}\,A,\ \mathrm{F}\,B)$, for some $\nabla \mathrel{\dot{\subseteq}} \Gamma$. The following derivation in $\mathbf{C}_{inv}$ proves this case:

$$\dfrac{\dfrac{\dfrac{\nabla,\ \mathrm{T}\,A,\ \mathrm{F}\,B}{\nabla,\ \mathrm{F}\,A \supset B,\ \mathrm{F}\,B}\ (\mathrm{F} \supset_l)}{\nabla,\ \mathrm{F}\,A \supset B,\ \mathrm{F}\,A \supset B}\ (\mathrm{F} \supset_r)}{\nabla,\ \mathrm{F}\,A \supset B}\ (C).$$

This case also demonstrates the need for the contraction rule in $\mathbf{C}_{inv}$.

SUBCASE $(\mathrm{T}\,A \mathrel{\dot{\in}} \Delta$ and $\mathrm{F}\,B \mathrel{\dot{\notin}} \Delta))$. We have $\Delta = (\nabla,\ \mathrm{T}\,A)$, for some $\nabla \mathrel{\dot{\subseteq}} \Gamma$. The following derivation in $\mathbf{C}_{inv}$ proves this case:

$$\frac{\nabla,\ \mathrm{T}\,A}{\nabla,\ \mathrm{F}\,A \supset B}\ (\mathrm{F} \supset_l).$$

SUBCASE $(\mathrm{F}\,B \mathrel{\dot{\in}} \Delta$ and $\mathrm{T}\,A \mathrel{\dot{\notin}} \Delta))$. This case is similar to the previous one.

SUBCASE $(\mathrm{T}\,A \mathrel{\dot{\notin}} \Delta$ and $\mathrm{F}\,B \mathrel{\dot{\notin}} \Delta)$. In this case we let $\Delta' = \Delta$.

CASE $(\mathrm{T} \supset)$. The rule of $\mathbf{C}_{tab}$ has the form

$$\frac{\Gamma,\ \mathrm{F}\,A \quad \Gamma,\ \mathrm{T}\,B}{\Gamma,\ \mathrm{T}\,A \supset B}\ (\mathrm{T} \supset).$$

By the induction hypothesis, there exist sequents $\Delta_1 \mathrel{\dot{\subseteq}} (\Gamma,\ \mathrm{F}\,A)$ and $\Delta_2 \mathrel{\dot{\subseteq}} (\Gamma,\ \mathrm{T}\,B)$ that have refutations in $\mathbf{C}_{inv}$. We have to find a sequent $\Delta \mathrel{\dot{\subseteq}} (\Gamma,\ \mathrm{T}\,A \supset B)$ refutable in $\mathbf{C}_{inv}$.
If we have $\Delta_1 \mathrel{\dot{\subseteq}} \Gamma$, then we can take Δ_1 to be Δ. Likewise, if we have $\Delta_2 \mathrel{\dot{\subseteq}} \Gamma$, then we can take Δ_2 to be Δ. Therefore, we can assume that $\Delta_1 = (\Delta_1',\ \mathrm{F}\,A)$ and $\Delta_2 = (\Delta_2',\ \mathrm{T}\,B)$ for some $\Delta_1' \mathrel{\dot{\subseteq}} \Gamma$ and $\Delta_2' \mathrel{\dot{\subseteq}} \Gamma$. Since $\Delta_1' \mathrel{\dot{\subseteq}} \Gamma$ and $\Delta_2' \mathrel{\dot{\subseteq}} \Gamma$, some sequent $\Gamma' \mathrel{\dot{\subseteq}} \Gamma$ can be obtained from the sequent Δ_1', Δ_2' in $\mathbf{C}_{inv}$ by a series of contractions. The following derivation in $\mathbf{C}_{inv}$ proves this case:

$$\begin{array}{c} \dfrac{\Delta_1',\ \mathrm{F}\,A \quad \Delta_2',\ \mathrm{T}\,B}{\Delta_1',\ \Delta_2',\ \mathrm{T}\,A \supset B}\ (\mathrm{T} \supset) \\ \vdots\ \text{series of contractions} \\ \Gamma',\ \mathrm{T}\,A \supset B \end{array}$$

□

The name *subsequent lemma* is due to the fact that we prove the existence of a subsequent Δ of Γ.

Using this lemma, we obtain completeness of $\mathbf{C}_{inv}$ for formulas:

3.8. THEOREM (Completeness of $\mathbf{C}_{inv}$). *Let ξ be a closed signed formula. ξ is unsatisfiable if and only if it has a refutation in $\mathbf{C}_{inv}$.*

PROOF. By the Completeness Theorem 3.6 for $\mathbf{C}_{tab}$, ξ is unsatisfiable if and only if there exists a refutation of ξ in $\mathbf{C}_{tab}$. By Subsequent Lemma 3.7, there exists such a refutation of ξ if and only if some submultiset of ξ has a refutation in $\mathbf{C}_{inv}$. Since the empty sequent has no refutation in $\mathbf{C}_{inv}$, the only submultiset of ξ refutable in $\mathbf{C}_{inv}$ is ξ itself. Therefore, ξ is unsatisfiable if and only if it has a refutation in $\mathbf{C}_{inv}$. □

3.9. EXAMPLE. Consider the signed formula

$$\xi = \mathrm{F}\,\forall y \exists x \exists z (P(z,x) \supset P(z,fx) \wedge P(x,ffy)). \tag{3.1}$$

(We omit parentheses in terms like $f(f(y))$ in this and other examples.) A refutation of ξ in $\mathbf{C}_{tab}$ is shown in Figure 4. Applying the algorithm implicit in the proof of the Subsequent Lemma to this derivation, we obtain the derivation of (3.1) in $\mathbf{C}_{inv}$ shown in Figure 5.

3.2. Subformula property of $\mathbf{C}_{inv}$

Following the universal recipe of automated deduction, we have obtained a suitable ground version $\mathbf{C}_{inv}$ of a sequent calculus intended to be used for a forward proof-search. The next step is to obtain a free-variable version of $\mathbf{C}_{inv}$. In the free-variable version, automatic refutations of a goal formula G will be obtained by repeatedly applying inference rules of the free-variable calculus in the forward direction to initial sequents (axioms). Evidently, it is desirable to restrict ourselves to a finite number of axioms. To this end, we will use the subformula property of $\mathbf{C}_{inv}$: a derivation of a signed formula ξ may only involve signed subformulas of ξ.

There are two standard definitions of subformulas. The first one defines subformulas of $\exists x A(x)$ as $A(x)$ plus subformulas of $A(x)$. The second one used e.g. in [Gentzen 1934], defines subformulas of $\exists x A(x)$ as all formulas of the form $A(t)$, where t is a term, plus subformulas of $A(t)$. We use both definitions, but to distinguish between them we call subformulas of the first kind *free subformulas*. In order to give the right definition of a signed subformula, it is enough to analyze the rules of $\mathbf{C}_{inv}$ (or, alternatively, the rules of $\mathbf{C}_{tab}$).

3.10. DEFINITION *(signed subformula)*. The notion of a ***free signed subformula*** of a signed formula α is defined by induction as follows. First, we define ***free immediate signed subformulas*** and ***immediate signed subformulas*** using the table of Figure 6.

$$
\begin{array}{cl}
\mathrm{T}\,P(fy,ffy),\ \mathrm{F}\,P(fy,ffy),\ldots \qquad \mathrm{T}\,P(fy,ffy),\ \mathrm{F}\,P(fy,ffy),\ldots & \\
\hline
\mathrm{T}\,P(fy,ffy),\ \mathrm{F}\,P(fy,ffy)\wedge P(fy,ffy),\ldots & (\mathrm{F}\wedge) \\
\hline
\mathrm{T}\,P(fy,ffy),\ \mathrm{F}\,P(fy,fy)\supset P(fy,ffy)\wedge P(fy,ffy),\ldots & (\mathrm{F}\supset) \\
\hline
\mathrm{T}\,P(fy,ffy),\ \mathrm{F}\,\exists z(P(z,fy)\supset P(z,ffy)\wedge P(fy,ffy)),\ldots & (\mathrm{F}\exists) \\
\hline
\mathrm{T}\,P(fy,ffy),\ \mathrm{F}\,\exists x\exists z(P(z,x)\supset P(z,fx)\wedge P(x,ffy)),\ldots & (\mathrm{F}\exists) \\
\hline
\mathrm{F}\,P(fy,ffy)\supset P(fy,fffy)\wedge P(ffy,ffy), & (\mathrm{F}\supset) \\
\mathrm{F}\,\exists x\exists z(P(z,x)\supset P(z,fx)\wedge P(x,ffy)),\ldots & \\
\hline
\mathrm{F}\,\exists z(P(z,ffy)\supset P(z,fffy)\wedge P(ffy,ffy)), & (\mathrm{F}\exists) \\
\mathrm{F}\,\exists x\exists z(P(z,x)\supset P(z,fx)\wedge P(x,ffy)) & \\
\hline
\mathrm{F}\,\exists x\exists z(P(z,x)\supset P(z,fx)\wedge P(x,ffy)) & (\mathrm{F}\exists) \\
\hline
\mathrm{F}\,\forall y\exists x\exists z(P(z,x)\supset P(z,fx)\wedge P(x,ffy)) & (\mathrm{F}\forall)
\end{array}
$$

In this derivation some irrelevant formulas are replaced by "...". For example, in the top sequents "..." denotes the multiset

$$
\begin{array}{l}
\mathrm{F}\,\exists x\exists z(P(z,x)\supset P(z,fx)\wedge P(x,ffy)),\\
\mathrm{F}\,\exists z(P(z,ffy)\supset P(z,fffy)\wedge P(ffy,ffy)),\\
\mathrm{F}\,\exists z(P(z,fy)\supset P(z,ffy)\wedge P(fy,ffy)),\\
\mathrm{F}\,P(fy,fffy)\wedge P(ffy,ffy),\\
\mathrm{T}\,P(fy,fy).
\end{array}
$$

Figure 4: $\mathbf{C}_{tab}$-derivation

The following $\mathbf{C}_{inv}$-derivation is obtained from the derivation above as in the subsequent lemma.

$$
\begin{array}{cl}
\mathrm{T}\,P(fy,ffy),\ \mathrm{F}\,P(fy,ffy) \qquad \mathrm{T}\,P(fy,ffy),\ \mathrm{F}\,P(fy,ffy) & \\
\hline
\mathrm{T}\,P(fy,ffy),\ \mathrm{T}\,P(fy,ffy),\ \mathrm{F}\,P(fy,ffy)\wedge P(fy,ffy) & (\mathrm{F}\wedge) \\
\hline
\mathrm{T}\,P(fy,ffy),\ \mathrm{F}\,P(fy,ffy)\wedge P(fy,ffy) & (C) \\
\hline
\mathrm{T}\,P(fy,ffy),\ \mathrm{F}\,P(fy,fy)\supset P(fy,ffy)\wedge P(fy,ffy) & (\mathrm{F}\supset_r) \\
\hline
\mathrm{T}\,P(fy,ffy),\ \mathrm{F}\,\exists z(P(z,fy)\supset P(z,ffy)\wedge P(fy,ffy)) & (\mathrm{F}\exists) \\
\hline
\mathrm{T}\,P(fy,ffy),\ \mathrm{F}\,\exists x\exists z(P(z,x)\supset P(z,fx)\wedge P(x,ffy)) & (\mathrm{F}\exists) \\
\hline
\mathrm{F}\,P(fy,ffy)\supset P(fy,fffy)\wedge P(ffy,ffy), & (\mathrm{F}\supset_l) \\
\mathrm{F}\,\exists x\exists z(P(z,x)\supset P(z,fx)\wedge P(x,ffy)) & \\
\hline
\mathrm{F}\,\exists z(P(z,ffy)\supset P(z,fffy)\wedge P(ffy,ffy)), & (\mathrm{F}\exists) \\
\mathrm{F}\,\exists x\exists z(P(z,x)\supset P(z,fx)\wedge P(x,ffy)) & \\
\hline
\mathrm{F}\,\exists x\exists z(P(z,x)\supset P(z,fx)\wedge P(x,ffy)), & (\mathrm{F}\exists) \\
\mathrm{F}\,\exists x\exists z(P(z,x)\supset P(z,fx)\wedge P(x,ffy)) & \\
\hline
\mathrm{F}\,\exists x\exists z(P(z,x)\supset P(z,fx)\wedge P(x,ffy)) & (C) \\
\hline
\mathrm{F}\,\forall y\exists x\exists z(P(z,x)\supset P(z,fx)\wedge P(x,ffy)) & (\mathrm{F}\forall)
\end{array}
$$

Figure 5: The corresponding $\mathbf{C}_{inv}$-derivation

In the table below, t is an arbitrary term free for x in $A(x)$.

signed formula	its free immediate signed subformulas	its immediate signed subformulas
$\mathrm{T}\, A \wedge B$	$\mathrm{T}\, A$ and $\mathrm{T}\, B$	$\mathrm{T}\, A$ and $\mathrm{T}\, B$
$\mathrm{F}\, A \wedge B$	$\mathrm{F}\, A$ and $\mathrm{F}\, B$	$\mathrm{F}\, A$ and $\mathrm{F}\, B$
$\mathrm{T}\, A \vee B$	$\mathrm{T}\, A$ and $\mathrm{T}\, B$	$\mathrm{T}\, A$ and $\mathrm{T}\, B$
$\mathrm{F}\, A \vee B$	$\mathrm{F}\, A$ and $\mathrm{F}\, B$	$\mathrm{F}\, A$ and $\mathrm{F}\, B$
$\mathrm{T}\, A \supset B$	$\mathrm{F}\, A$ and $\mathrm{T}\, B$	$\mathrm{F}\, A$ and $\mathrm{T}\, B$
$\mathrm{F}\, A \supset B$	$\mathrm{T}\, A$ and $\mathrm{F}\, B$	$\mathrm{T}\, A$ and $\mathrm{F}\, B$
$\mathrm{T}\, \neg A$	$\mathrm{F}\, A$	$\mathrm{F}\, A$
$\mathrm{F}\, \neg A$	$\mathrm{T}\, A$	$\mathrm{T}\, A$
$\mathrm{T}\, \exists x A(x)$	$\mathrm{T}\, A(x)$	$\mathrm{T}\, A(t)$
$\mathrm{F}\, \exists x A(x)$	$\mathrm{F}\, A(x)$	$\mathrm{F}\, A(t)$
$\mathrm{T}\, \forall x A(x)$	$\mathrm{T}\, A(x)$	$\mathrm{T}\, A(t)$
$\mathrm{F}\, \forall x A(x)$	$\mathrm{F}\, A(x)$	$\mathrm{F}\, A(t)$

Figure 6: Immediate signed subformulas

For all signed formulas α, β, γ we define various notions of subformulas as follows.

1. α is a (free) signed subformula of α;
2. If α is a (free) immediate signed subformula of β and β is a (free) signed subformula of γ, then α is a (free) signed subformula of γ.

We already mentioned the ***subformula property*** several times without defining it. There can be several definitions of the subformula property. Below we give several forms of subformula property, the first is the strongest.

- *Every derivation* of a sequent Γ consists of signed subformulas of signed formulas in Γ;
- *Every refutation* of a sequent Γ consists of signed subformulas of signed formulas in Γ;
- If Γ has a refutation, then *there exists* a refutation of Γ consisting of signed subformulas of signed formulas in Γ.
- Every *derivation* of a sequent Γ consists of subformulas of signed formulas in Γ.

All free-variable calculi discussed in this chapter have the fourth form of the subformula property. All other calculi of this chapter have the first, strongest form.

3.11. Theorem (Subformula Property of $\mathbf{C}_{inv}$). *Let Π be a derivation of a signed formula ξ in $\mathbf{C}_{inv}$. Every signed formula occurring in Π is a signed subformula of ξ.*

PROOF. By a routine inspection of inference rules of $\mathbf{C}_{inv}$. □

Thus, when we search for a refutation of a particular signed formula ξ, we can restrict our search to sequents consisting of signed subformulas of ξ. When ξ contains quantifiers, it may have an infinite number of signed subformulas, so the subformula property does not restrict the search space good enough. The next thing to note is that signed subformulas of ξ have a convenient representation in terms of *free* signed subformulas, and any signed formula has only a finite number of free signed subformulas.

3.2.1. Representation of subformulas by free subformulas

We call a (signed) formula α ***rectified*** if (i) every occurrence of a quantifier in α binds a different variable and (ii) every bound variable of α has no free occurrence in α. Evidently, we can make any (signed) formula into an equivalent rectified one by renaming of bound variables. A sequent Γ is ***rectified*** if so is every signed formula in Γ.

CONVENTION $\boxed{\text{In the sequel we assume all formulas to be rectified.}}$

We will often represent signed subformulas of a given formula ξ in the form $\alpha \cdot \theta$, where α is a free signed subformula of ξ and θ is a substitution admissible for A. We say that a pair $\alpha \cdot \theta$ ***represents*** the signed subformula $\alpha\theta$. We also call this representation the representation ***via a free signed subformula***. Note that the same signed subformula β can have different representations via different free signed subformulas.

We call a substitution θ ***relevant*** to an expression E if $dom(\theta) \subseteq free(E)$. We introduce a convention of a technical nature. Whenever we use the notation $\alpha \cdot \theta$ such that θ is not relevant to α, we mean the representation $\alpha \cdot \tau$, where τ is the restriction of θ on $free(\alpha)$. This convention will considerably simplify notation.

3.12. LEMMA. *Let α, β be rectified signed formulas. Then β is a signed subformula of α if and only if $\beta = \gamma\theta$ for some free signed subformula γ of α and substitution θ such that $dom(\theta) \cap free(\alpha) = \emptyset$. For given β and γ such a substitution θ is unique among substitutions relevant to γ.*

PROOF. By routine inspection of the definition of signed subformulas. □

In particular, every signed subformula of a closed signed formula can be obtained from a free signed subformula by applying a substitution. Now we can reformulate the subformula property of $\mathbf{C}_{inv}$ in the following way.

3.13. COROLLARY. *Let Π be a refutation of a rectified closed signed formula ξ in $\mathbf{C}_{inv}$. Every signed formula occurring in Π has the form $\gamma\theta$ for a free signed subformula γ of ξ and a substitution θ.* □

Suppose that we want to check the refutability in $\mathbf{C}_{inv}$ of a closed signed formula ξ. By Corollary 3.13, we can restrict signed formulas occurring in the derivation to signed formulas of the form $\gamma\theta$, where γ is a free signed subformula of ξ and θ is a substitution. Since this applies to axioms as well, every axiom has the form $\mathrm{T}\,A\theta,\ \mathrm{F}\,B\sigma$, where A, B are atomic and $\mathrm{T}\,A,\ \mathrm{F}\,B$ are *free* signed subformulas of ξ. For any given A, B, there may be an infinite number of such axioms because of different choices for substitutions θ, σ, but there is only a *finite* number of pairs of *free* signed subformulas $\mathrm{T}\,A$ and $\mathrm{F}\,B$ of Σ. As usual in automated deduction, we can choose a *most general* axiom that represents all axioms $\mathrm{T}\,A\theta,\ \mathrm{F}\,B\sigma$ for given A and B.

3.3. The calculus $\mathbf{C}^{\xi}_{inv}$ *and lifting*

We will now introduce the main calculus $\mathbf{C}^{\xi}_{inv}$ for the inverse method for classical logic. The calculus is based on the idea of representing sequents through free subformulas and using most general unifiers instead of arbitrary substitutions.

As usual, we assume that ξ is a rectified closed signed formula (the goal). A ***sequent of*** $\mathbf{C}^{\xi}_{inv}$ is any multiset of pairs $\alpha \cdot \sigma$ that represent subformulas of ξ. We say that a sequent Γ of $\mathbf{C}_{inv}$ is an ***instance*** of a sequent $\alpha_1 \cdot \sigma_1, \ldots, \alpha_n \cdot \sigma_n$ of $\mathbf{C}^{\xi}_{inv}$ if there exists a substitution τ such that the multiset $\alpha_1\sigma_1\tau, \ldots, \alpha_n\sigma_n\tau$ coincides with Γ.

Now our aim is the following. Given a closed signed formula ξ, to make a sequent calculus $\mathbf{C}^{\xi}_{inv}$ that has the following property: for every sequent Γ refutable in $\mathbf{C}_{inv}$ and consisting of signed subformulas of ξ, there exists a sequent Δ refutable in $\mathbf{C}^{\xi}_{inv}$ such that Γ is an instance of Δ. Before defining such a calculus we have to introduce more definitions.

A ***restriction*** of a substitution σ to a set of variables V, denoted by $\sigma\restriction_V$, is the substitution defined as follows. For every variable x

$$x(\sigma\restriction_V) \rightleftharpoons \begin{cases} x\sigma, & \text{if } x \in V; \\ x, & \text{otherwise.} \end{cases}$$

We denote by σ_{-x} the substitution $\sigma\restriction_{dom(\sigma)\setminus\{x\}}$. It is easy to see that

$$y\sigma_{-x} = \begin{cases} x, & \text{if } x = y; \\ y\sigma, & \text{otherwise.} \end{cases}$$

A substitution σ is ***more general*** than a substitution θ, denoted $\sigma \leq \theta$ if there exists a substitution τ such that $\sigma\tau = \theta$.

We introduce the notions related to *unification* of expressions and substitutions.

3.14. DEFINITION *(unification).* Let E_1, E_2 be two expressions and θ a substitution admissible for E_1, E_2. We call θ a ***unifier*** of E_1 and E_2 if $E_1\theta = E_2\theta$. Two expressions are ***unifiable*** if they have a unifier. A unifier θ of E_1 and E_2 is called

most general, denoted $\mathrm{mgu}(E_1, E_2)$, if θ is more general than any unifier of E_1 and E_2.

A ***most general unifier of substitutions*** σ and τ, denoted $\mathrm{mgu}(\sigma, \tau)$, is a substitution θ defined as follows. Let $\{x_1, \ldots, x_n\} = dom(\sigma) \cup dom(\tau)$, $n \geq 0$. Then θ is any most general unifier of the tuples $\langle x_1\sigma, \ldots, x_n\sigma\rangle$ and $\langle x_1\tau, \ldots, x_n\tau\rangle$.

A substitution ρ is called a ***renaming*** if ρ is a one-to-one mapping from its domain to itself. It is easy to see that a substitution ρ is a renaming if and only if $dom(\rho) = ran(\rho)$. Also ρ is a renaming if and only if there exists a substitution τ such that $\rho\tau = \varepsilon$ (or $\tau\rho = \varepsilon$). Let us note, that if a substitution σ is a most general unifier of terms t and s, then $\sigma\rho$ is also a most general unifier of these terms for any renaming ρ.

3.15. DEFINITION *(free variables, variants, renaming).* Let $\Gamma = \alpha_1 \cdot \sigma_1, \ldots, \alpha_n \cdot \sigma_n$. The set of ***free variables*** of Γ, denoted by $\mathit{free}(\Gamma)$, is the set of all free variables of $\alpha_1\sigma_1, \ldots, \alpha_n\sigma_n$. For a substitution θ we denote by $\Gamma\theta$ the sequent $\alpha_1 \cdot \sigma_1\theta, \ldots, \alpha_n \cdot \sigma_n\theta$. We call a ***variant*** of Γ any sequent $\Gamma\rho$, where ρ is a renaming. Two expressions E_1 and E_2 are ***variable-disjoint*** if $\mathit{free}(E_1) \cap \mathit{free}(E_2) = \emptyset$. We say that a substitution ρ ***renames*** E_1 ***away from*** E_2 if ρ is a renaming such that $E_1\rho$ and E_2 are variable-disjoint. If, for such a renaming, $E_1\rho$ and E_2 are unifiable, we say that E_1 and E_2 are ***weakly unifiable***.

Like in standard resolution calculi, we identify sequents that are variants of each other. Therefore, we assume an implicit ***renaming rule***:

$$\frac{\Gamma}{\Gamma\rho}\ \text{, where } \rho \text{ is a renaming.}$$

The calculus $\mathbf{C}^{\xi}_{inv}$ is given in Figure 7. An example derivation in $\mathbf{C}^{\xi}_{inv}$, when ξ is (3.1), is shown in Figure 8. Compare this derivation with that of Figure 5 on page 193.

We can prove completeness of $\mathbf{C}^{\xi}_{inv}$ by comparing $\mathbf{C}_{inv}$ with $\mathbf{C}^{\xi}_{inv}$ and using lifting arguments, expressed by the following lemma.

3.16. LEMMA (Instance Lemma for $\mathbf{C}^{\xi}_{inv}$). *Let ξ be a closed signed formula and let $\alpha_1, \ldots, \alpha_n$ be free signed subformulas of ξ. Let $\theta_1, \ldots, \theta_n$ be substitutions such that the sequent $\alpha_1\theta_1, \ldots, \alpha_n\theta_n$ is derivable in $\mathbf{C}_{inv}$. Then there exist substitutions $\lambda, \tau_1, \ldots, \tau_n$ such that the sequent $\alpha_1 \cdot \tau_1, \ldots, \alpha_n \cdot \tau_n$ is derivable in $\mathbf{C}^{\xi}_{inv}$ and $\alpha_i\tau_i\lambda = \alpha_i\theta_i$ for all $i \in \{1, \ldots, n\}$.*

PROOF. The proof is by induction on the length of a derivation Π in $\mathbf{C}_{inv}$. We will consider the last of inference of Π and assume, by induction hypothesis, that the lemma holds for all the premises of this inference. We consider several cases, depending on the last inference, the other cases will be similar. These cases are the following:

- the last rule of Π is (Ax);

All signed formulas in the inference rules are signed subformulas of ξ. In the rule (Ax) below, A and B are weakly unifiable atoms, ρ renames away A from B and τ is a most general unifier of $A\rho$ and B. In all rules the premises are variable-disjoint. In all rules each premise is also variable-disjoint with $var(\xi)$. In all rules θ is a most general unifier of σ_1, σ_2. The rules $(\mathrm{F}\,\forall)$ and $(\mathrm{T}\,\exists)$ satisfy the eigenvariable condition: the term $x\sigma$ is a variable not occurring in the conclusion of the rule.

$$\frac{}{\mathrm{T}\,A \cdot \rho\tau, \mathrm{F}\,B \cdot \tau}\ (Ax)$$

$$\frac{\Gamma,\ \gamma \cdot \sigma_1,\ \gamma \cdot \sigma_2}{\Gamma\theta,\ \gamma \cdot \sigma_1\theta}\ (C)$$

$$\frac{\Gamma,\ \mathrm{T}\,A \cdot \sigma}{\Gamma,\ \mathrm{T}\,A \wedge B \cdot \sigma}\ (\mathrm{T} \wedge_l) \qquad \frac{\Gamma,\ \mathrm{T}\,A \cdot \sigma}{\Gamma,\ \mathrm{T}\,B \wedge A \cdot \sigma}\ (\mathrm{T} \wedge_r)$$

$$\frac{\Gamma,\ \mathrm{F}\,A \cdot \sigma_1 \quad \Delta,\ \mathrm{F}\,B \cdot \sigma_2}{\Gamma\theta,\ \Delta\theta,\ \mathrm{F}\,A \wedge B \cdot \sigma_1\theta}\ (\mathrm{F} \wedge)$$

$$\frac{\Gamma,\ \mathrm{T}\,A \cdot \sigma_1 \quad \Delta,\ \mathrm{T}\,B \cdot \sigma_2}{\Gamma\theta,\ \Delta\theta,\ \mathrm{T}\,A \vee B \cdot \sigma_1\theta}\ (\mathrm{T} \vee)$$

$$\frac{\Gamma,\ \mathrm{F}\,A \cdot \sigma}{\Gamma,\ \mathrm{F}\,A \vee B \cdot \sigma}\ (\mathrm{F} \vee_l) \qquad \frac{\Gamma,\ \mathrm{F}\,B \cdot \sigma}{\Gamma,\ \mathrm{F}\,A \vee B \cdot \sigma}\ (\mathrm{F} \vee_r)$$

$$\frac{\Gamma,\ \mathrm{F}\,A \cdot \sigma_1 \quad \Delta,\ \mathrm{T}\,B \cdot \sigma_2}{\Gamma\theta,\ \Delta\theta,\ \mathrm{T}\,A \supset B \cdot \sigma_1\theta}\ (\mathrm{T} \supset)$$

$$\frac{\Gamma,\ \mathrm{T}\,A \cdot \sigma}{\Gamma,\ \mathrm{F}\,A \supset B \cdot \sigma}\ (\mathrm{F} \supset_l) \qquad \frac{\Gamma,\ \mathrm{F}\,B \cdot \sigma}{\Gamma,\ \mathrm{F}\,A \supset B \cdot \sigma}\ (\mathrm{F} \supset_r)$$

$$\frac{\Gamma,\ \mathrm{F}\,A \cdot \sigma}{\Gamma,\ \mathrm{T}\,\neg A \cdot \sigma}\ (\mathrm{T} \neg) \qquad \frac{\Gamma,\ \mathrm{T}\,A \cdot \sigma}{\Gamma,\ \mathrm{F}\,\neg A \cdot \sigma}\ (\mathrm{F} \neg)$$

$$\frac{\Gamma,\ \mathrm{T}\,A \cdot \sigma}{\Gamma,\ \mathrm{T}\,\forall x A \cdot \sigma_{-x}}\ (\mathrm{T} \forall) \qquad \frac{\Gamma,\ \mathrm{F}\,A \cdot \sigma}{\Gamma,\ \mathrm{F}\,\forall x A \cdot \sigma_{-x}}\ (\mathrm{F} \forall)$$

$$\frac{\Gamma,\ \mathrm{T}\,A \cdot \sigma}{\Gamma,\ \mathrm{T}\,\exists x A \cdot \sigma_{-x}}\ (\mathrm{T} \exists) \qquad \frac{\Gamma,\ \mathrm{F}\,A \cdot \sigma}{\Gamma,\ \mathrm{F}\,\exists x A \cdot \sigma_{-x}}\ (\mathrm{F} \exists)$$

Figure 7: Calculus $\mathbf{C}^{\xi}_{inv}$

$$\dfrac{\mathrm{T}\,P(z,x)\cdot\{x\mapsto fx_2, z\mapsto x_1\}, \mathrm{F}\,P(z,fx)\cdot\{x\mapsto x_2, z\mapsto x_1\} \qquad \mathrm{T}\,P(z,x)\cdot\{x\mapsto ffy_2, z\mapsto y_1\}, \mathrm{F}\,P(x,ffy)\cdot\{x\mapsto y_1, y\mapsto y_2\}}{\mathrm{T}\,P(z,x)\cdot\{x\mapsto fx_2, z\mapsto x_1\}, \mathrm{T}\,P(z,x)\cdot\{x\mapsto ffx_3, z\mapsto x_2\}, \mathrm{F}\,P(z,fx)\wedge P(x,ffy)\cdot\{x\mapsto x_2, y\mapsto x_3, z\mapsto x_1\}}\ (\mathrm{F}\wedge)$$

$$\dfrac{\mathrm{T}\,P(z,x)\cdot\{x\mapsto fx_2, z\mapsto x_1\}, \mathrm{T}\,P(z,x)\cdot\{x\mapsto ffx_3, z\mapsto x_2\}, \mathrm{F}\,P(z,fx)\wedge P(x,ffy)\cdot\{x\mapsto x_2, y\mapsto x_3, z\mapsto x_1\}}{\mathrm{T}\,P(z,x)\cdot\{x\mapsto ffx_1, z\mapsto fx_1\}, \mathrm{F}\,P(z,fx)\wedge P(x,ffy)\cdot\{x\mapsto fx_1, y\mapsto x_1, z\mapsto fx_1\}}\ (C)$$

$$\dfrac{\mathrm{T}\,P(z,x)\cdot\{x\mapsto ffx_1, z\mapsto fx_1\}, \mathrm{F}\,P(z,fx)\wedge P(x,ffy)\cdot\{x\mapsto fx_1, y\mapsto x_1, z\mapsto fx_1\}}{\mathrm{T}\,P(z,x)\cdot\{x\mapsto ffx_1, z\mapsto fx_1\}, \mathrm{F}\,P(z,x)\supset P(z,fx)\wedge P(x,ffy)\cdot\{x\mapsto fx_1, y\mapsto x_1, z\mapsto fx_1\}}\ (\mathrm{F}\ \supset_r)$$

$$\dfrac{\mathrm{T}\,P(z,x)\cdot\{x\mapsto ffx_1, z\mapsto fx_1\}, \mathrm{F}\,P(z,x)\supset P(z,fx)\wedge P(x,ffy)\cdot\{x\mapsto fx_1, y\mapsto x_1, z\mapsto fx_1\}}{\mathrm{T}\,P(z,x)\cdot\{x\mapsto ffx_1, z\mapsto fx_1\}, \mathrm{F}\,\exists z(P(z,x)\supset P(z,fx)\wedge P(x,ffy))\cdot\{x\mapsto fx_1, y\mapsto x_1\}}\ (\mathrm{F}\,\exists)$$

$$\dfrac{\mathrm{T}\,P(z,x)\cdot\{x\mapsto ffx_1, z\mapsto fx_1\}, \mathrm{F}\,\exists z(P(z,x)\supset P(z,fx)\wedge P(x,ffy))\cdot\{x\mapsto fx_1, y\mapsto x_1\}}{\mathrm{T}\,P(z,x)\cdot\{x\mapsto ffx_1, z\mapsto fx_1\}, \mathrm{F}\,\exists x\exists z(P(z,x)\supset P(z,fx)\wedge P(x,ffy))\cdot\{y\mapsto x_1\}}\ (\mathrm{F}\,\exists)$$

$$\dfrac{\mathrm{T}\,P(z,x)\cdot\{x\mapsto ffx_1, z\mapsto fx_1\}, \mathrm{F}\,\exists x\exists z(P(z,x)\supset P(z,fx)\wedge P(x,ffy))\cdot\{y\mapsto x_1\}}{\mathrm{F}\,P(z,x)\supset P(z,fx)\wedge P(x,ffy)\cdot\{x\mapsto ffx_1, y\mapsto x_2, z\mapsto fx_1\}, \mathrm{F}\,\exists x\exists z(P(z,x)\supset P(z,fx)\wedge P(x,ffy))\cdot\{y\mapsto x_1\}}\ (\mathrm{F}\ \supset_l)$$

$$\dfrac{\mathrm{F}\,P(z,x)\supset P(z,fx)\wedge P(x,ffy)\cdot\{x\mapsto ffx_1, y\mapsto x_2, z\mapsto fx_1\}, \mathrm{F}\,\exists x\exists z(P(z,x)\supset P(z,fx)\wedge P(x,ffy))\cdot\{y\mapsto x_1\}}{\mathrm{F}\,\exists z(P(z,x)\supset P(z,fx)\wedge P(x,ffy))\cdot\{x\mapsto ffx_1, y\mapsto x_2\}, \mathrm{F}\,\exists x\exists z(P(z,x)\supset P(z,fx)\wedge P(x,ffy))\cdot\{y\mapsto x_1\}}\ (\mathrm{F}\,\exists)$$

$$\dfrac{\mathrm{F}\,\exists z(P(z,x)\supset P(z,fx)\wedge P(x,ffy))\cdot\{x\mapsto ffx_1, y\mapsto x_2\}, \mathrm{F}\,\exists x\exists z(P(z,x)\supset P(z,fx)\wedge P(x,ffy))\cdot\{y\mapsto x_1\}}{\mathrm{F}\,\exists x\exists z(P(z,x)\supset P(z,fx)\wedge P(x,ffy))\cdot\{y\mapsto x_2\}, \mathrm{F}\,\exists x\exists z(P(z,x)\supset P(z,fx)\wedge P(x,ffy))\cdot\{y\mapsto x_1\}}\ (\mathrm{F}\,\exists)$$

$$\dfrac{\mathrm{F}\,\exists x\exists z(P(z,x)\supset P(z,fx)\wedge P(x,ffy))\cdot\{y\mapsto x_2\}, \mathrm{F}\,\exists x\exists z(P(z,x)\supset P(z,fx)\wedge P(x,ffy))\cdot\{y\mapsto x_1\}}{\mathrm{F}\,\exists x\exists z(P(z,x)\supset P(z,fx)\wedge P(x,ffy))\cdot\{y\mapsto x_1\}}\ (C)$$

$$\dfrac{\mathrm{F}\,\exists x\exists z(P(z,x)\supset P(z,fx)\wedge P(x,ffy))\cdot\{y\mapsto x_1\}}{\mathrm{F}\,\forall y\exists x\exists z(P(z,x)\supset P(z,fx)\wedge P(x,ffy))\cdot\varepsilon}\ (\mathrm{F}\,\forall)$$

Figure 8: An $\mathbf{C}^{\xi}_{inv}$-derivation

- the last rule of Π is (C);
- the last rule of Π is $(\mathrm{F}\wedge)$;
- the last rule of Π is $(\mathrm{F}\vee_l)$;
- the last rule of Π is $(\mathrm{F}\forall)$;
- the last rule of Π is $(\mathrm{F}\exists)$.

At the beginning we note the following useful property of sequents in $\mathbf{C}^{\xi}_{inv}$-derivations.

> For every rule of $\mathbf{C}^{\xi}_{inv}$ if a pair $\alpha \cdot \tau$ is in the premise then $dom(\tau) = free(\alpha)$. Furthermore we can suppose that the same property holds for all pairs $\alpha \cdot \tau$ in sequents derived by induction hypothesis because of the renaming rule. (3.2)

Indeed, $dom(\tau) \subseteq free(\alpha)$ because we assume the substitution τ to be relevant for α. Likewise, $free(\alpha) \subseteq dom(\tau)$ because we require the premises to be variable-disjoint from $var(\xi)$, and therefore, from $free(\alpha)$.

In order to simplify the presentation we suppose that every sequent Γ in the rules of $\mathbf{C}_{inv}$, except for (Ax) and (C) consists only of one formula, and the sequents in (Ax) and (C) consist of at most two formulas.

CASE (Ax). In this case the last inference has the form

$$\frac{}{\mathrm{T}\,A_1\theta_1, \mathrm{F}\,A_2\theta_2}\ (Ax),$$

where $\mathrm{T}\,A_1, \mathrm{F}\,A_2$ are free signed subformulas of ξ and $A_1\theta_1 = A_2\theta_2$. Without loss of generality we can assume that $dom(\theta_1) \subseteq free(A_1)$ $(dom(\theta_2) \subseteq free(A_2))$, otherwise we can take the substitutions $\theta_j\lceil_{free(A_j)}$ instead of θ_j, for $j = 1, 2$. Let a renaming ρ rename A_1 away from A_2. Since $A_1\rho(\rho^{-1}\theta_1) = A_2\theta_2$ we infer that $((\rho^{-1}\theta_1)\lceil_{free(A_1\rho)} \cup \theta_2)$ is a unifier of $A_1\rho$ and A_2. Therefore there exists a most general unifier σ of $A_1\rho$ and A_2 such that $((\rho^{-1}\theta_1)\lceil_{free(A_1\rho)} \cup \theta_2) = \sigma\lambda$ for a substitution λ. We can conclude that the figure

$$\frac{}{\mathrm{T}\,A_1 \cdot \rho\sigma, \mathrm{F}\,A_2 \cdot \sigma}\ (Ax)$$

is the intended refutation in $\mathbf{C}^{\xi}_{inv}$. Indeed,

$$A_1\theta_1 = A_1\rho((\rho^{-1}\theta_1)\lceil_{free(A_1\rho)} \cup \theta_2) = A_1\rho\sigma\lambda,$$
$$A_2\theta_2 = A_2((\rho^{-1}\theta_1)\lceil_{free(A_1\rho)} \cup \theta_2) = A_2\sigma\lambda.$$

CASE (C). The last inference in the $\mathbf{C}_{inv}$-refutation is

$$\frac{\alpha_1\theta_1, \alpha_2\theta_2}{\alpha_1\theta_1}\ (C),$$

such that $\alpha_1\theta_1 = \alpha_2\theta_2$. By the induction hypothesis, there exist substitutions τ_1, τ_2, λ' such that $\alpha_j\theta_j = \alpha_j\tau_j\lambda'$ and the sequents $\alpha_j \cdot \tau_j$ are derivable in $\mathbf{C}^{\xi}_{inv}$, for

$j = 1, 2$. This implies that λ' is a unifier of $\alpha_1\theta_1$ and $\alpha_2\theta_2$. Therefore, there exists a most general unifier σ of $\alpha_1\theta_1$ and $\alpha_2\theta_2$ and a substitution λ such that $\sigma\lambda = \lambda'$. The following inference in $\mathbf{C}^{\xi}_{inv}$ completes the proof of this case

$$\frac{\alpha_1 \cdot \tau_1, \alpha_2 \cdot \tau_2}{\alpha_1 \cdot \tau_1\sigma}\ (C).$$

CASE (F $\wedge$). The last inference in the $\mathbf{C}_{inv}$-refutation is

$$\frac{\mathrm{F}\, A_1\theta_1 \quad \mathrm{F}\, A_2\theta_2}{\mathrm{F}\,(A_1 \wedge A_2)\theta}\ (\mathrm{F}\wedge),$$

where, $\theta_j = \theta\lceil_{free(A_j)}$, for $j = 1, 2$. By the induction hypothesis, there are sequents $\mathrm{F}\, A_j \cdot \tau_j$ derivable in $\mathbf{C}^{\xi}_{inv}$ and the substitutions λ_j such that $\mathrm{F}\, A_j\theta_j = \alpha_j\tau_j\lambda_j$, for $j = 1, 2$. By using the renaming rule, we can assume $dom(\lambda_1) \cap dom(\lambda_1) = \emptyset$.
Let $\{x_1, \ldots, x_n\} = dom(\tau_1)\setminus dom(\tau_2)$ and $\{y_1, \ldots, y_m\} = dom(\tau_2)\setminus dom(\tau_1)$. Let $\lambda'_1 = \lambda_1 \cup \{y_1 \mapsto y_1\tau_2\lambda_2, \ldots, y_m \mapsto y_m\tau_2\lambda_2\}$ and $\lambda'_2 = \lambda_2 \cup \{x_1 \mapsto x_1\tau_1\lambda_1, \ldots, x_n \mapsto x_n\tau_1\lambda_1\}$. We can define a new substitution $\lambda' = \lambda'_1 \cup \lambda'_2$ and put it instead of λ_1 and λ_2 in the previous two equations. It results in the conclusion that λ' is a unifier of τ_1 and τ_2 because for every $x \in (dom(\tau_1) \cup dom(\tau_2))$ it holds that $x\tau_1\lambda' = x\tau_2\lambda' = x\theta$. Let $\sigma = \mathrm{mgu}(\tau_1, \tau_2)$, then $\lambda' = \sigma\lambda$ for some substitution λ.
The following inference completes the proof of this case:

$$\frac{\mathrm{F}\, A_1 \cdot \tau_1 \quad \mathrm{F}\, A_2 \cdot \tau_2}{\mathrm{F}\, A_1 \wedge A_2 \cdot \tau_1\sigma}\ (\mathrm{F}\wedge).$$

CASE (F $\vee_l$). The last inference in the $\mathbf{C}_{inv}$-refutation is

$$\frac{\mathrm{F}\, A\theta_1}{\mathrm{F}\,(A_1 \vee A_2)\theta}\ (\mathrm{F}\vee_l)$$

where $\theta_1 = \theta\lceil_{free(A_1)}$.
By the induction hypothesis, there exist a sequent $\mathrm{F}\, A_1 \cdot \tau_1$ refutable in $\mathbf{C}^{\xi}_{inv}$ and a substitution λ' such that $\mathrm{F}\, A_1\theta_1 = \mathrm{F}\, A\tau_1\lambda'$. Let us show that the following inference completes the proof of this case:

$$\frac{\mathrm{F}\, A_1 \cdot \tau_1}{\mathrm{F}\, A_1 \vee A_2 \cdot \tau_1}\ (\mathrm{F}\vee_l).$$

Let $\{x_1, \ldots, x_n\} = dom(\theta) \setminus free(A)$. Define $\lambda = \lambda' \cup \{x_1 \mapsto x_1\theta, \ldots x_n \mapsto x_n\theta\}$. Using $vran(\tau_1) \cap var(A_2) = \emptyset$, it is not hard to argue that $(\mathrm{F}\, A_1 \vee A_2)\tau\lambda = (\mathrm{F}\, A_1 \vee A_2)\theta$.
CASE (F $\forall$). The last inference of the $\mathbf{C}_{inv}$-refutation is

$$\frac{\mathrm{F}\, A(y) \cdot \theta}{\mathrm{F}\, \forall x A(x) \cdot \theta}\ (\mathrm{F}\forall). \tag{3.3}$$

For simplicity, we only consider the case when $x \in \mathit{free}(A)$, the other case can be proved in a similar way. In this inference $x, y \notin \mathit{dom}(\theta)$, therefore the inference can be rewritten as

$$\frac{\mathrm{F}\, A(x) \cdot \theta \cup \{x \mapsto y\}}{\mathrm{F}\, \forall x A(x) \cdot \theta}\ (\mathrm{F}\,\forall).$$

By the induction hypothesis, there exist substitutions τ, λ' and a sequent $\mathrm{F}\, A(x) \cdot \tau$ refutable in $\mathbf{C}^{\xi}_{inv}$ such that

$$A(x)\tau\lambda' = A(x)\theta \cup \{x \mapsto y\}. \tag{3.4}$$

To conclude the proof of this case, it is enough to find a substitution λ such that

$$\forall x A(x)\tau_{-x}\lambda = \forall x A(x)\theta \tag{3.5}$$

and

$$\frac{\mathrm{F}\, A(x) \cdot \tau}{\mathrm{F}\, \forall x A(x) \cdot \tau_{-x}}\ (\mathrm{F}\,\forall) \tag{3.6}$$

is a valid inference in $\mathbf{C}^{\xi}_{inv}$.
We prove (3.5) by letting $\lambda = \lambda'$. Indeed, since $x \notin \mathit{free}(\forall x A(x))$, we have $\forall x A(x)\tau_{-x}\lambda = \forall x A(x)\tau\lambda$. Condition (3.4) implies $\forall x A(x)\tau_{-x}\lambda = \forall x A(x)\theta \cup \{x \mapsto y\}$. Using $x \notin \mathit{dom}(\theta)$, we obtain (3.5).
To prove that (3.6) is a valid inference in $\mathbf{C}^{\xi}_{inv}$, it is enough to check the eigenvariable condition: $x\tau$ is a variable not occurring in $\forall x A(x)\tau_{-x}$. Condition (3.4) implies $x\tau\lambda' = x\theta \cup \{x \mapsto y\}$. Since $x \notin \mathit{dom}(\theta)$, we have $x\tau\lambda' = y$. Therefore, $x\tau$ is a variable, denote this variable by z. Note that $z\lambda' = y$. To check the eigenvariable condition suppose, by contradiction, $z \in \mathit{free}(\forall x A(x)\tau_{-x})$. Then $z\lambda' \in \mathit{free}(\forall x A(x)\tau_{-x}\lambda')$. From $z\lambda' = y$ it follows that $y \in \mathit{free}(\forall x A(x)\tau_{-x}\lambda')$. Equation (3.5) yields $y \in \mathit{free}(\forall x A(x)\theta)$. This contradicts the eigenvariable condition for inference (3.3).

CASE ($\mathrm{T}\,\forall$). This case is quite similar to the previous one, except that we do not need to verify the troublesome eigenvariable condition.

□

To establish the soundness of $\mathbf{C}^{\xi}_{inv}$, one can prove the following lemma.

3.17. LEMMA. *If a sequent $\alpha_1 \cdot \theta_1, \ldots, \alpha_n \cdot \theta_n$ has a refutation in $\mathbf{C}^{\xi}_{inv}$, then the sequent $\alpha_1\theta_1, \ldots, \alpha_n\theta_n$ has a refutation in $\mathbf{C}_{inv}$.* □

3.18. THEOREM (Completeness of $\mathbf{C}^{\xi}_{inv}$). *Let ξ be a closed signed formula. Then ξ is unsatisfiable if and only if ξ has a refutation in $\mathbf{C}^{\xi}_{inv}$.*

Proof. ($\Rightarrow$). Suppose that ξ is unsatisfiable. By Completeness Theorem 3.18 for $\mathbf{C}_{inv}$, ξ has a refutation in $\mathbf{C}_{inv}$. By Instance Lemma 3.16, ξ has a refutation in $\mathbf{C}^{\xi}_{inv}$.

($\Leftarrow$). Suppose that ξ has a refutation in $\mathbf{C}^{\xi}_{inv}$. By Lemma 3.17, ξ has a refutation in $\mathbf{C}_{inv}$. By Completeness Theorem 3.18 for $\mathbf{C}_{inv}$, ξ is unsatisfiable. □

It follows from this theorem that, for any closed formula A, the sequent $\mathrm{F}\,A$ has a refutation in $\mathbf{C}^{\xi}_{inv}$ if and only if A is valid.

Using unifiable atoms instead of weakly unifiable in the axiom (Ax) gives an incomplete calculus. Consider, for example, the signed formula $\xi = \mathrm{F}\,\exists x(P(x) \supset P(fx))$. To apply the rule (Ax) to $P(x)$ and $P(fx)$, we take $\rho = \{x \mapsto y\}$ and use the most general unifier $\{y \mapsto fx\}$ of $P(y)$ and $P(fx)$:

$$\mathrm{T}\,P(x) \cdot \{x \mapsto fx\},\ \mathrm{F}\,P(fx) \cdot \varepsilon.$$

This axiom can be extended to a refutation of the sequent. However, if we use unifiability instead of weak unifiability, no axiom can be built since $P(x)$ and $P(fx)$ are not unifiable.[1]

3.4. Negation normal forms

In order to decrease the number of rules and simplify the presentation, we will consider formulas of a special form, called the negation normal form. In classical logic and all modal logics discussed in this paper every formula is equivalent to a formula in negation normal form.

3.19. Definition *(literal, negation normal form).* A ***literal*** is an atomic formula or its negation. For every literal L, the literal ***complementary*** to L, denoted $\overline{L}$ is defined as follows:

$$\overline{L} = \begin{cases} A, & \text{if } L = \neg A; \\ \neg L, & \text{otherwise.} \end{cases}$$

A formula is in ***negation normal form***, or simply in ***NNF***, if it is constructed from literals using $\wedge$, $\vee$, $\forall$ and $\exists$. A *negation normal form of a formula A* is any formula in negation normal form equivalent to A.

We will now change the definition of signed subformulas into a definition of subformulas of NNFs.

3.20. Definition *(subformula).* The notion of a ***free subformula*** of a NNF A is defined by induction as follows. First, we define ***free immediate subformulas*** and ***immediate subformulas*** using the table of Figure 6.

[1] Tammet [1997] uses unifiability instead of weak unifiability in the axioms for intuitionistic logic. It seems that he assumes an implicit variable renaming rule. For intuitionistic logic, a similar counterexample to completeness is $\neg\neg\exists x(P(x) \supset P(fx))$.

In the table below, t is an arbitrary term free for x in $A(x)$.

formula	its free immediate subformulas	its immediate subformulas
$A \wedge B$	A and B	A and B
$A \vee B$	A and B	A and B
$\exists x A(x)$	$A(x)$	$A(t)$
$\forall x A(x)$	$A(x)$	$A(t)$

Figure 9: Immediate subformulas

For all formulas A, B, C we define various notions of subformulas as follows.

1. A is a (free) signed subformula of A;
2. If A is a (free) immediate subformula of B and B is a (free) subformula of C, then A is a (free) subformula of C.

Note that a formula $\neg A$ has no immediate subformulas, and therefore the only subformula of $\neg A$ is $\neg A$ itself.

It is known that every formula can be transformed in its negation normal form using the following transformations:

$$\begin{array}{rcl} A \supset B & \Rightarrow & \neg A \vee B, \\ \neg(A \vee B) & \Rightarrow & \neg A \wedge \neg B, \\ \neg(A \wedge B) & \Rightarrow & \neg A \vee \neg B, \\ \neg\neg A & \Rightarrow & A, \\ \neg \exists x A & \Rightarrow & \forall x \neg A, \\ \neg \forall x A & \Rightarrow & \exists x \neg A. \end{array}$$

When we have a formula A in negation normal form and try to refute $\mathrm{T}\, A$, the calculus $\mathbf{C}_{inv}$ can be considerably simplified. Essentially, we should only be dealing with signed formulas of the form $\mathrm{T}\, B$, where B is in negation normal form. Since there is only one sign, there is no reason to consider signs at all, so sequents can be viewed as multisets of formulas. A multiset of formulas $B_1, \ldots, B_n$ corresponds to the multiset of signed formulas $\mathrm{T}\, B_1, \ldots, \mathrm{T}\, B_n$. Instead of a goal signed formula $\xi = \mathrm{T}\, G$, we will deal with the goal formula G.

CONVENTION $\boxed{\text{In the sequel } G \text{ denotes the goal formula in NNF.}}$

The calculi $\mathbf{C}_{tab}$, $\mathbf{C}_{inv}$ and $\mathbf{C}^{\xi}_{inv}$ specialized to formulas in NNF are shown in Figures 10–12. Since we now deal with a goal formula G instead of a goal signed formula ξ, the calculus $\mathbf{C}^{\xi}_{inv}$ is denoted by $\mathbf{C}^{G}_{inv}$. We do not give conditions on the inference rules, the reader can derive these conditions from the conditions on the original calculi.

$$\frac{}{\Gamma,\ A,\ \neg A}\ (Ax) \qquad \frac{\Gamma,\ A \quad \Gamma,\ B}{\Gamma,\ A \vee B}\ (\vee) \qquad \frac{\Gamma,\ A,\ B}{\Gamma,\ A \wedge B}\ (\wedge)$$

$$\frac{\Gamma,\ A(y)}{\Gamma,\ \exists x A(x)}\ (\exists) \qquad \frac{\Gamma,\ \forall x A(x),\ A(t)}{\Gamma,\ \forall x A(x)}\ (\forall)$$

Figure 10: Calculus $\mathbf{C}_{tab}$ for formulas in NNF

$$\frac{}{A, \neg A}\ (Ax) \qquad \frac{\Gamma,\ A,\ A}{\Gamma,\ A}\ (C)$$

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

$$\frac{\Gamma,\ A(y)}{\Gamma,\ \exists x A(x)}\ (\exists) \qquad \frac{\Gamma,\ A(t)}{\Gamma,\ \forall x A(x)}\ (\forall)$$

Figure 11: Calculus $\mathbf{C}_{inv}$ for formulas in NNF

$$\frac{}{A \cdot \rho\tau, \neg B \cdot \tau}\ (Ax) \qquad \frac{\Gamma,\ A \cdot \sigma_1,\ A \cdot \sigma_2}{\Gamma\theta,\ A \cdot \sigma_1\theta}\ (C)$$

$$\frac{\Gamma,\ A \cdot \sigma_1 \quad \Delta,\ B \cdot \sigma_2}{\Gamma\theta,\ \Delta\theta,\ A \vee B \cdot \sigma_1\theta}\ (\vee)$$

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

$$\frac{\Gamma,\ A \cdot \sigma}{\Gamma,\ \exists x A \cdot \sigma_{-x}}\ (\exists) \qquad \frac{\Gamma,\ A \cdot \sigma}{\Gamma,\ \forall x A \cdot \sigma_{-x}}\ (\forall)$$

Figure 12: Calculus $\mathbf{C}^G_{inv}$ for formulas in NNF

3.21. EXAMPLE. Consider again signed formula (3.1) on page 192. The corresponding formula in negation normal form is

$$G = \exists y \forall x \forall z (P(z,x) \wedge (\neg P(z,fx) \vee \neg P(x,ffy))). \tag{3.7}$$

Using the calculus $\mathbf{C}^G_{inv}$ for formulas in negation normal norm, the derivation of (3.1) of Figure 8 can be turned into the derivation of this formula shown in Figure 13.

3.5. Saturation algorithms and linear representation of derivations

Proof-search by the inverse method is based on a ***saturation algorithm***. This algorithm makes iterative computations over a database of sequents. Initially, the database contains all axioms of $\mathbf{C}^G_{inv}$. At every iteration step an inference or set of inferences is applied to one or more sequents in the database. The conclusions of these inferences are added to the database. The proof-search terminates if the goal is in the database, or if no new sequents can be added. Sometimes, ***redundancy criteria*** are applied to the database. Each redundancy criterion identifies sequents that are redundant and can be removed from the database. There are also redundancy criteria for inferences. Redundancy criteria for the inverse method are discussed in Section 6. Saturation-based algorithms and redundancies for resolution are discussed in [Bachmair and Ganzinger 2001] (Chapter 2 of this Handbook), for paramodulation-based reasoning in [Nieuwenhuis and Rubio 2001] (Chapter 7), and from the practical viewpoint in [Weidenbach 2001] (Chapter 27).

In such saturation algorithms, the same sequents can be reused over and over again. Typical derivation trees can be tremendously large (in resolution-based inference systems more than 2^{50} clauses), while the number of sequents used in the derivation is typically small (no more than a few thousand). Therefore, we obtain a more economical representation of derivations by only displaying the sequents instead of derivation trees. We will call them ***linear representations of derivations***. [2]

In the sequel, we will only use linear representations of derivations, with numbered sequents. With every sequent, we will write the name of the inference rule used to obtain this sequent, and the numbers of its parents: the sequents in the premises of the inference used to obtain this sequent. The linear representation of the derivation of Figure 13 is shown in Figure 14.

3.6. Exploiting other properties of sequent calculi

What we have achieved in this section is the construction of the calculus $\mathbf{C}^G_{inv}$ that can be used to derive formulas in an automatic way. To make a further step towards proof-search, we have to introduce some *redundancy criteria* to restrict the

[2] A *linear derivation* would be an appropriate term for such kind of derivations, however, this term resembles linear formats for resolution (see e.g., [Kowalski and Kuehner 1971]).

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{P(z,x)\cdot\{x\mapsto fx_2, z\mapsto x_1\}, \neg P(z,fx)\cdot\{x\mapsto x_2, z\mapsto x_1\} \qquad P(z,x)\cdot\{x\mapsto ffy_2, z\mapsto y_1\}, \neg P(x,ffy)\cdot\{x\mapsto y_1, y\mapsto y_2\}}{P(z,x)\cdot\{x\mapsto fx_2, z\mapsto x_1\}, P(z,x)\cdot\{x\mapsto ffx_3, z\mapsto x_2\}, \neg P(z,fx)\vee\neg P(x,ffy)\cdot\{x\mapsto x_2, y\mapsto x_3, z\mapsto x_1\}}\;(\vee)}{P(z,x)\cdot\{x\mapsto ffx_1, z\mapsto fx_1\}, \neg P(z,fx)\vee\neg P(x,ffy)\cdot\{x\mapsto fx_1, y\mapsto x_1, z\mapsto fx_1\}}\;(C)}{P(z,x)\cdot\{x\mapsto ffx_1, z\mapsto fx_1\}, P(z,x)\wedge(\neg P(z,fx)\vee\neg P(x,ffy))\cdot\{x\mapsto fx_1, y\mapsto x_1, z\mapsto fx_1\}}\;(\wedge_r)}{P(z,x)\cdot\{x\mapsto ffx_1, z\mapsto fx_1\}, \forall z(P(z,x)\wedge(\neg P(z,fx)\vee\neg P(x,ffy)))\cdot\{x\mapsto fx_1, y\mapsto x_1\}}\;(\forall)}{P(z,x)\cdot\{x\mapsto ffx_1, z\mapsto fx_1\}, \forall x\forall z(P(z,x)\wedge(\neg P(z,fx)\vee\neg P(x,ffy)))\cdot\{y\mapsto x_1\}}\;(\forall)}{P(z,x)\wedge(\neg P(z,fx)\vee\neg P(x,ffy))\cdot\{x\mapsto ffx_1, y\mapsto x_2, z\mapsto fx_1\}, \forall x\forall z(P(z,x)\wedge(\neg P(z,fx)\vee\neg P(x,ffy)))\cdot\{y\mapsto x_1\}}\;(\wedge_l)}{\forall z(P(z,x)\wedge(\neg P(z,fx)\vee\neg P(x,ffy)))\cdot\{x\mapsto ffx_1, y\mapsto x_2\}, \forall x\forall z(P(z,x)\wedge(\neg P(z,fx)\vee\neg P(x,ffy)))\cdot\{y\mapsto x_1\}}\;(\forall)}{\forall x\forall z(P(z,x)\wedge(\neg P(z,fx)\vee\neg P(x,ffy)))\cdot\{y\mapsto x_2\}, \forall x\forall z(P(z,x)\wedge(\neg P(z,fx)\vee\neg P(x,ffy)))\cdot\{y\mapsto x_1\}}\;(\forall)}{\forall x\forall z(P(z,x)\wedge\neg(P(z,fx)\vee\neg P(x,ffy)))\cdot\{y\mapsto x_1\}}\;(C)}{\exists y\forall x\forall z(P(z,x)\wedge(\neg P(z,fx)\vee\neg P(x,ffy)))\cdot\varepsilon}\;(\exists)
$$

Figure 13: An $\mathbf{C}^G_{inv}$-derivation of a formula in negation normal form

(1)	$P(z,x) \cdot \{x \mapsto fx_2, z \mapsto x_1\}$,	
	$\neg P(z,fx) \cdot \{x \mapsto x_2, z \mapsto x_1\}$	axiom
(2)	$P(z,x) \cdot \{x \mapsto ffy_2, z \mapsto y_1\}$,	
	$\neg P(x,ffy) \cdot \{x \mapsto y_1, y \mapsto y_2\}$	axiom
(3)	$P(z,x) \cdot \{x \mapsto fx_2, z \mapsto x_1\}$,	
	$P(z,x) \cdot \{x \mapsto ffx_3, z \mapsto x_2\}$,	
	$\neg P(z,fx) \vee \neg P(x,ffy) \cdot \{x \mapsto x_2, y \mapsto x_3, z \mapsto x_1\}$	$(\vee)$ from (1,2)
(4)	$P(z,x) \cdot \{x \mapsto ffx_1, z \mapsto fx_1\}$,	
	$\neg P(z,fx) \vee \neg P(x,ffy) \cdot \{x \mapsto fx_1, y \mapsto x_1, z \mapsto fx_1\}$	(C) from (3)
(5)	$P(z,x) \cdot \{x \mapsto ffx_1, z \mapsto fx_1\}$,	
	$P(z,x) \wedge (\neg P(z,fx) \vee \neg P(x,ffy))$	
	$\cdot\{x \mapsto fx_1, y \mapsto x_1, z \mapsto fx_1\}$	$(\wedge_r)$ from (4)
(6)	$P(z,x) \cdot \{x \mapsto ffx_1, z \mapsto fx_1\}$,	
	$\forall z(P(z,x) \wedge (\neg P(z,fx) \vee \neg P(x,ffy))) \cdot \{x \mapsto fx_1, y \mapsto x_1\}$	$(\forall)$ from (5)
(7)	$P(z,x) \cdot \{x \mapsto ffx_1, z \mapsto fx_1\}$,	
	$\forall x \forall z(P(z,x) \wedge (\neg P(z,fx) \vee \neg P(x,ffy))) \cdot \{y \mapsto x_1\}$	$(\forall)$ from (6)
(8)	$P(z,x) \wedge (\neg P(z,fx) \vee \neg P(x,ffy))$	
	$\cdot\{x \mapsto ffx_1, y \mapsto x_2, z \mapsto fx_1\}$,	
	$\forall x \forall z(P(z,x) \wedge (\neg P(z,fx) \vee \neg P(x,ffy))) \cdot \{y \mapsto x_1\}$	$(\wedge_l)$ from (7)
(9)	$\forall z(P(z,x) \wedge (\neg P(z,fx) \vee \neg P(x,ffy))) \cdot \{x \mapsto ffx_1, y \mapsto x_2\}$,	
	$\forall x \forall z(P(z,x) \wedge (\neg P(z,fx) \vee \neg P(x,ffy))) \cdot \{y \mapsto x_1\}$	$(\forall)$ from (8)
(10)	$\forall x \forall z(P(z,x) \wedge (\neg P(z,fx) \vee \neg P(x,ffy))) \cdot \{y \mapsto x_2\}$,	
	$\forall x \forall z(P(z,x) \wedge (\neg P(z,fx) \vee \neg P(x,ffy))) \cdot \{y \mapsto x_1\}$	$(\forall)$ from (9)
(11)	$\forall x \forall z(P(z,x) \wedge (\neg P(z,fx) \vee \neg P(x,ffy))) \cdot \{y \mapsto x_1\}$	(C) from (10)
(12)	$\exists y \forall x \forall z(P(z,x) \wedge (\neg P(z,fx) \vee \neg P(x,ffy))) \cdot \varepsilon$	$(\exists)$ from (11)

Figure 14: The linear representation of the derivation of Figure 13

search space, similar to those extensively studied in [Bachmair and Ganzinger 2001] (Chapter 2 of this Handbook). There are essentially two kinds of redundancies in proof-search systems: redundant sequents and redundant derivations. For the inverse method, some of these criteria (like subsumption) can be taken over from resolution or from other very general concepts [Maslov 1983*c*, Voronkov 1992, Mints, Orevkov and Tammet 1996]. Other criteria are based on properties of sequent calculi. We will consider redundancy criteria in Section 6.

Another important aspect is the representation of sequents. In Section 5 we will consider a naming technique that allows one to consider clauses instead of sequents. In the case of classical logic, the naming technique augmented with the treatment of inference rules as clauses gives us the simulation of the inverse method by hyperresolution [Maslov 1969, Maslov 1971*b*, Kuechner 1971], see also [Bachmair and

Ganzinger 2001, page 76] (Chapter 2 of this volume).

4. Applying the recipe to nonclassical logics

In this section we apply the universal recipe to cook the inverse method calculi for several nonclassical logics: intuitionistic logic and some modal logics. There are only a few papers on the inverse method for nonclassical logics, and some authors characterize the inverse method as "resolution" for nonclassical logic due to the close relationships between resolution derivations and the inverse method derivations explained in Section 5.2. We avoid calling the inverse method "resolution" for the following reasons.

1. The inverse method is not regarded as resolution for classical logic, where there is a bisimulation of hyperresolution derivations and the inverse method derivations.
2. The inverse method systems for nonclassical logics contain inference rules that cannot be simulated by the standard resolution rule. As a consequence, the "resolution" calculi for nonclassical logics contain "clauses" encoding these rules for which special non-resolution inference rules are introduced. There is no semantics behind these rules unless we translate them back to the sequent calculus rules.

The inverse method for nonclassical logics is discussed in [Mints 1990, Auffray, Enjalbert and Hebrard 1990, Mints et al. 1996, Voronkov 2000*a*, Voronkov 2001, Voronkov 2000*b*] for modal logics, in [Voronkov 1992] for a number of logics, including intuitionistic logic, modal logics and logics without the contraction rule, in [Mints 1993*b*, Tammet 1994] for linear logic, and in [Mints 1994, Tammet 1996, Tammet 1997] for intuitionistic logic.

4.1. Intuitionistic logic

In the case of intuitionistic logic, the design of a suitable inverse method calculus follows the same recipe as for classical logic. We begin with a cut-free calculus $\mathbf{I}_{tab}$, change it into an inverse method calculus, and finally use the subformula property to add free variables to it.

We call an ***intuitionistic sequent*** any sequent with at most one signed formula of the form $\mathrm{F}\, A$. The *sequent calculus for intuitionistic logic* $\mathbf{I}_{tab}$ is the calculus of intuitionistic sequents shown in Figure 15. This calculus is essentially $G3$ of Kleene [1967, page 481].

To find the right rules for the inverse method calculus out of $\mathbf{I}_{tab}$, we have to adapt the proof of Subsequent Lemma 3.7 to the intuitionistic case. This gives us the calculus $\mathbf{I}_{inv}$ shown in Figure 16. Let us explain why some rules of the system look different from their classical counterpart, even when the tableau rules are the same. A typical example is the rule $(\mathrm{T}\vee)$. For classical logic, the inverse method rule has the form

$$\frac{}{\Gamma,\ \mathrm{T}\,A,\ \mathrm{F}\,A}\ (Ax)$$

$$\frac{\Gamma,\ \mathrm{T}\,A,\ \mathrm{T}\,B}{\Gamma,\ \mathrm{T}\,A \wedge B}\ (\mathrm{T}\wedge) \qquad \frac{\Gamma,\ \mathrm{F}\,A \quad \Gamma,\ \mathrm{F}\,B}{\Gamma,\ \mathrm{F}\,A \wedge B}\ (\mathrm{F}\wedge)$$

$$\frac{\Gamma,\ \mathrm{T}\,A \quad \Gamma,\ \mathrm{T}\,B}{\Gamma,\ \mathrm{T}\,A \vee B}\ (\mathrm{T}\vee) \qquad \frac{\Gamma,\ \mathrm{F}\,A}{\Gamma,\ \mathrm{F}\,A \vee B}\ (\mathrm{F}\vee_l) \qquad \frac{\Gamma,\ \mathrm{F}\,B}{\Gamma,\ \mathrm{F}\,A \vee B}\ (\mathrm{F}\vee_r)$$

$$\frac{\Gamma,\ \mathrm{T}\,A \supset B,\ \mathrm{F}\,A \quad \Gamma,\ \mathrm{T}\,B}{\Gamma,\ \mathrm{T}\,A \supset B}\ (\mathrm{T}\supset) \qquad \frac{\Gamma,\ \mathrm{T}\,A,\ \mathrm{F}\,B}{\Gamma,\ \mathrm{F}\,A \supset B}\ (\mathrm{F}\supset)$$

$$\frac{\Gamma,\ \mathrm{T}\,\neg A,\ \mathrm{F}\,A}{\Gamma,\ \mathrm{T}\,\neg A}\ (\mathrm{T}\neg_1) \qquad \frac{\Gamma,\ \mathrm{T}\,\neg A,\ \mathrm{F}\,A}{\Gamma,\ \mathrm{T}\,\neg A,\ \mathrm{F}\,B}\ (\mathrm{T}\neg_2) \qquad \frac{\Gamma,\ \mathrm{T}\,A}{\Gamma,\ \mathrm{F}\,\neg A}\ (\mathrm{F}\neg)$$

$$\frac{\Gamma,\ \mathrm{T}\,\forall x A(x),\ \mathrm{T}\,A(t)}{\Gamma,\ \mathrm{T}\,\forall x A(x)}\ (\mathrm{T}\forall) \qquad \frac{\Gamma,\ \mathrm{F}\,A(y)}{\Gamma,\ \mathrm{F}\,\forall x A(x)}\ (\mathrm{F}\forall)$$

$$\frac{\Gamma,\ \mathrm{T}\,A(y)}{\Gamma,\ \mathrm{T}\,\exists x A(x)}\ (\mathrm{T}\exists) \qquad \frac{\Gamma,\ \mathrm{F}\,A(t)}{\Gamma,\ \mathrm{F}\,\exists x A(x)}\ (\mathrm{F}\exists)$$

In the rule (Ax), A is an atomic formula. The rules $(\mathrm{F}\forall)$ and $(\mathrm{T}\exists)$ satisfy the eigenvariable condition.

Figure 15: Calculus $\mathbf{I}_{tab}$

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

In order to obtain the sequent $\mathrm{T}\,A\vee B,\ \mathrm{F}\,C$ from $\mathrm{T}\,A,\ \mathrm{F}\,C$ and $\mathrm{T}\,B\ ,\mathrm{F}\,C$ in classical logic, we can use the following derivation

$$\cfrac{\cfrac{\mathrm{T}\,A,\ \mathrm{F}\,C \quad \mathrm{T}\,B,\ \mathrm{F}\,C}{\mathrm{T}\,A \vee B,\ \mathrm{F}\,C,\ \mathrm{F}\,C}\ (\mathrm{T}\vee)}{\mathrm{T}\,A \vee B,\ \mathrm{F}\,C}\ (C). \tag{4.1}$$

In the intuitionistic system, this derivation cannot be used, since $\mathrm{T}\,A\vee B,\ \mathrm{F}\,C,\ \mathrm{F}\,C$ is not a valid sequent. To avoid this problem, we augment the intuitionistic inverse method system $\mathbf{I}_{inv}$ with an additional rule

$$\frac{\Gamma,\ \mathrm{T}\,A,\ \mathrm{F}\,C \quad \Delta,\ \mathrm{T}\,B,\ \mathrm{F}\,C}{\Gamma,\ \Delta,\ \mathrm{T}\,A \vee B,\ \mathrm{F}\,C}\ (\mathrm{T}\vee_2).$$

The problem we had with the rule $(\mathrm{T}\vee)$ appears due to the impossibility of applying contraction to a signed formula $(\mathrm{F}\,C)$: such an application would require two

$$\frac{}{\mathrm{T}\,A, \mathrm{F}\,A}\ (Ax)$$

$$\frac{\Gamma,\ \mathrm{T}\,A,\ \mathrm{T}\,A}{\Gamma,\ \mathrm{T}\,A}\ (C)$$

$$\frac{\Gamma,\ \mathrm{T}\,A}{\Gamma,\ \mathrm{T}\,A \wedge B}\ (\mathrm{T}\wedge) \qquad \frac{\Gamma,\ \mathrm{T}\,B}{\Gamma,\ \mathrm{T}\,A \wedge B}\ (\mathrm{T}\wedge)$$

$$\frac{\Gamma,\ \mathrm{F}\,A \quad \Delta,\ \mathrm{F}\,B}{\Gamma,\ \Delta,\ \mathrm{F}\,A \wedge B}\ (\mathrm{F}\wedge)$$

$$\frac{\Gamma,\ \mathrm{T}\,A \quad \Delta,\ \mathrm{T}\,B}{\Gamma,\ \Delta,\ \mathrm{T}\,A \vee B}\ (\mathrm{T}\vee_1) \qquad \frac{\Gamma,\ \mathrm{T}\,A, \mathrm{F}\,C \quad \Delta,\ \mathrm{T}\,B, \mathrm{F}\,C}{\Gamma,\ \Delta,\ \mathrm{T}\,A \vee B, \mathrm{F}\,C}\ (\mathrm{T}\vee_2)$$

$$\frac{\Gamma,\ \mathrm{F}\,A}{\Gamma,\ \mathrm{F}\,A \vee B}\ (\mathrm{F}\vee_l) \qquad \frac{\Gamma,\ \mathrm{F}\,B}{\Gamma,\ \mathrm{F}\,A \vee B}\ (\mathrm{F}\vee_r)$$

$$\frac{\Gamma,\ \mathrm{F}\,A \quad \Delta,\ \mathrm{T}\,B}{\Gamma,\ \Delta,\ \mathrm{T}\,A \supset B}\ (\mathrm{T} \supset)$$

$$\frac{\Gamma,\ \mathrm{T}\,A}{\Gamma,\ \mathrm{F}\,A \supset B}\ (\mathrm{F} \supset_l) \qquad \frac{\Gamma,\ \mathrm{F}\,B}{\Gamma,\ \mathrm{F}\,A \supset B}\ (\mathrm{F} \supset_r)$$

$$\frac{\Gamma,\ \mathrm{T}\,A,\ \mathrm{F}\,A \supset B}{\Gamma,\ \mathrm{F}\,A \supset B}\ (\mathrm{F} \supset_{lr})$$

$$\frac{\Gamma,\ \mathrm{F}\,A}{\Gamma,\ \mathrm{T}\,\neg A}\ (\mathrm{T}\,\neg) \qquad \frac{\Gamma,\ \mathrm{T}\,A}{\Gamma,\ \mathrm{F}\,\neg A}\ (\mathrm{F}\,\neg)$$

$$\frac{\Gamma,\ \mathrm{T}\,A(t)}{\Gamma,\ \mathrm{T}\,\forall x A(x)}\ (\mathrm{T}\,\forall) \qquad \frac{\Gamma,\ \mathrm{F}\,A(y)}{\Gamma,\ \mathrm{F}\,\forall x A(x)}\ (\mathrm{F}\,\forall)$$

$$\frac{\Gamma,\ \mathrm{T}\,A(y)}{\Gamma,\ \mathrm{T}\,\exists x A(x)}\ (\mathrm{T}\,\exists) \qquad \frac{\Gamma,\ \mathrm{F}\,A(t)}{\Gamma,\ \mathrm{F}\,\exists x A(x)}\ (\mathrm{F}\,\exists)$$

In the rule (Ax), A is an atomic formula. The rules $(\mathrm{F}\,\forall)$ and $(\mathrm{T}\,\exists)$ satisfy the eigenvariable condition.

Figure 16: Calculus $\mathbf{I}_{inv}$

copies of $(\mathrm{F}\,C)$ in a sequent, which is prohibited by the definition of intuitionistic sequents. The same problem arises in connection with the rule $(\mathrm{F}\ \supset)$. This problem was overlooked in [Tammet 1996, Tammet 1997] where an incorrect recipe for intuitionistic logic is given.

4.1. LEMMA. *Let Γ have a refutation in $\mathbf{I}_{tab}$. Then there exists a sequent Δ such that $\Delta \dot{\subseteq} \Gamma$ and Δ has a refutation in $\mathbf{I}_{inv}$.*

PROOF. The proof is similar to that of Lemma 3.7. We only consider some cases where the rules for intuitionistic logic are substantially different from those for classical logic. These are $(\mathrm{T}\ \supset)$, $(\mathrm{T}\,\neg)$ (the rules of $\mathbf{I}_{tab}$ are quite different from those of $\mathbf{C}_{tab}$) and $(\mathrm{F}\ \supset)$ (the rules of $\mathbf{I}_{inv}$ are quite different from those of $\mathbf{C}_{inv}$).
CASE $(\mathrm{T}\ \supset)$. The rule of $\mathbf{I}_{tab}$ has the form

$$\frac{\Gamma,\ \mathrm{T}\,A \supset B,\ \mathrm{F}\,A \quad \Gamma,\ \mathrm{T}\,B}{\Gamma,\ \mathrm{T}\,A \supset B}\ (\mathrm{T}\ \supset).$$

By the induction hypothesis, there exist sequents $\Delta_1 \dot{\subseteq} (\Gamma,\ \mathrm{T}\,A \supset B,\ \mathrm{F}\,A)$ and $\Delta_2 \dot{\subseteq} (\Gamma,\ \mathrm{T}\,B)$ is refutable in $\mathbf{I}_{inv}$. We have to find a sequent $\Delta \dot{\subseteq} (\Gamma,\ \mathrm{T}\,A \supset B)$ refutable in $\mathbf{I}_{inv}$.
If we have $\Delta_1 \dot{\subseteq} (\Gamma,\ \mathrm{T}\,A \supset B)$, then we can take Δ_1 to be Δ. Likewise, if we have $\Delta_2 \dot{\subseteq} \Gamma$, then we can take Δ_2 to be Δ. So we can assume that $(\mathrm{F}\,A) \in \Delta_1$ and $(\mathrm{T}\,B) \in \Delta_2$. We will only consider the case when $\mathrm{T}\,A \supset B$ occurs in Δ_1, leaving the remaining case to the reader. In this case we can assume that $\Delta_1 = (\Delta_1',\ \mathrm{T}\,A \supset B,\ \mathrm{F}\,A)$ and $\Delta_2 = (\Delta_2',\ \mathrm{T}\,B)$ for some $\Delta_1' \dot{\subseteq} \Gamma$ and $\Delta_2' \dot{\subseteq} \Gamma$. Since $\Delta_1' \dot{\subseteq} \Gamma$ and $\Delta_2' \dot{\subseteq} \Gamma$, and Γ contains no formulas of the form $\mathrm{F}\,C$, some sequent $\Gamma' \dot{\subseteq} \Gamma$ can be obtained from the sequent Δ_1', Δ_2' in $\mathbf{I}_{inv}$ by a series of contractions. The following derivation in $\mathbf{I}_{inv}$ proves this case:

$$\begin{array}{c}\dfrac{\Delta_1',\ \mathrm{T}\,A \supset B,\ \mathrm{F}\,A \quad \Delta_2',\ \mathrm{T}\,B}{\Delta_1',\ \Delta_2',\ \mathrm{T}\,A \supset B,\ \mathrm{T}\,A \supset B}\ (\mathrm{T}\ \supset) \\ \vdots\ \text{series of contractions} \\ \Gamma',\ \mathrm{T}\,A \supset B\end{array}$$

CASE $(\mathrm{F}\ \supset)$. This case gives rise to three rules of $\mathbf{I}_{inv}$. The $(\mathrm{F}\ \supset)$ rule of $\mathbf{I}_{tab}$ has the form

$$\frac{\Gamma,\ \mathrm{T}\,A,\ \mathrm{F}\,B}{\Gamma,\ \mathrm{F}\,A \supset B}\ (\mathrm{F}\ \supset).$$

By the induction hypothesis, there exists a sequent $\Delta \dot{\subseteq} (\Gamma,\ \mathrm{T}\,A,\ \mathrm{T}\,B)$ refutable in $\mathbf{I}_{inv}$. We have to find a sequent $\Delta' \dot{\subseteq} (\Gamma,\ \mathrm{F}\,A \supset B)$ refutable in $\mathbf{I}_{inv}$. Consider the following cases, depending on whether $\mathrm{T}\,A$ and $\mathrm{F}\,B$ occur in Δ.

SUBCASE $((\mathrm{T}\,A, \mathrm{F}\,B) \dot{\subseteq} \Delta)$. We have $\Delta = (\nabla,\ \mathrm{T}\,A,\ \mathrm{F}\,B)$, for some $\nabla \dot{\subseteq} \Gamma$. The following derivation in $\mathbf{I}_{inv}$ proves this case:

$$\cfrac{\cfrac{\nabla,\ \mathrm{T}\,A,\ \mathrm{F}\,B}{\nabla,\ \mathrm{T}\,A,\ \mathrm{F}\,A \supset B}\ (\mathrm{F}\ \supset_r)}{\nabla,\ \mathrm{F}\,A \supset B}\ (\mathrm{F}\ \supset_{lr}).$$

Subcase ($\mathrm{T}\,A \in \Delta$ and $\Delta \dot{\subseteq} (\Gamma,\ \mathrm{T}\,A)$). We have $\Delta = \nabla,\ \mathrm{T}\,A$, for some $\nabla \dot{\subseteq} \Gamma$. The following derivation in $\mathbf{I}_{inv}$ proves this case:

$$\frac{\nabla,\ \mathrm{T}\,A}{\nabla,\ \mathrm{F}\,A \supset B}\ (\mathrm{F} \supset_l).$$

Subcase ($\mathrm{T}\,B \in \Delta$ and $\Delta \dot{\subseteq} (\Gamma,\ \mathrm{T}\,B)$). This case is similar to the previous one.

Subcase ($\Delta \dot{\subseteq} \Gamma$). In this case we let $\Delta' = \Delta$.

Case ($\mathrm{T}\,\neg$). This case is similar to ($\mathrm{F}\ \supset$), but requires an extra contraction by ($\mathrm{T}\,\neg A$), like in the case ($\mathrm{T}\ \supset$). We leave the details to the reader.

□

As we can see from the proof, the application of one part of the recipe to intuitionistic logic gives us a few surprises. Two rules for ($\mathrm{T}\,\neg$) in $\mathbf{I}_{tab}$ become one rule in $\mathbf{I}_{inv}$, while one rule for ($\mathrm{F}\ \supset$) in $\mathbf{I}_{tab}$ becomes three rules in $\mathbf{I}_{inv}$. We could have obtained a smaller number of rules had we used sets instead of multisets. However, the use of sets would have caused a problem with lifting, resulting in the same free-variable system, and in a more complex proof of completeness for the free-variable system.

Before defining the calculus $\mathbf{I}^{\xi}_{inv}$ we have to generalize the notion of most general unifier of substitutions to two pairs of substitutions. A ***simultaneous most general unifier of the pairs of substitutions*** σ_1, τ_1 and σ_2, τ_2 is a substitution ϑ defined as follows. Let $\{x_1, \ldots, x_n\} = dom(\sigma_1) \cup dom(\sigma_2)$ and $\{y_1, \ldots, y_m\} = dom(\tau_1) \cup dom(\tau_2)$, $n, m \geq 0$. Then ϑ is any most general unifier of the tuples $\langle x_1\sigma_1, \ldots, x_n\sigma_1, y_1\tau_1, \ldots, y_m\tau_1\rangle$ and $\langle x_1\sigma_2, \ldots, x_n\sigma_2, y_1\tau_2, \ldots, y_m\tau_2\rangle$. The idea of this definition is that we have to unify σ_1 with σ_2 and also τ_1 with τ_2 *simultaneously*.

The free-variable system $\mathbf{I}^{\xi}_{inv}$ is given in Figure 17.

Again, by comparing $\mathbf{I}_{inv}$ and $\mathbf{I}^{\xi}_{inv}$ we obtain the same lifting arguments as for $\mathbf{C}_{inv}$ and $\mathbf{C}^{\xi}_{inv}$.

4.2. Lemma (Instance Lemma for $\mathbf{I}^{\xi}_{inv}$). *Let ξ be a closed signed formula and $\alpha_1, \ldots, \alpha_n$ be free signed subformulas of ξ. Let $\theta_1, \ldots, \theta_n$ be substitutions such that the sequent $\alpha_1\theta_1, \ldots, \alpha_n\theta_n$ is derivable in $\mathbf{I}_{inv}$. Then there exist substitutions $\lambda, \tau_1, \ldots, \tau_n$ such that the sequent $\alpha_1 \cdot \tau_1, \ldots, \alpha_n \cdot \tau_n$ is derivable in $\mathbf{I}^{\xi}_{inv}$ and $\alpha_i\tau_i\lambda = \alpha_i\theta_i$ for all $i \in \{1, \ldots, n\}$.*

Proof. The proof is quite similar to the proof of Lemma 3.16 above. In view of that proof, we will only consider the induction step in the case when the last inference of the derivation Π in $\mathbf{I}_{inv}$ is ($T\vee_2$). Derivation 4.1 on page 210 demonstrates that any inference by this rule in $\mathbf{I}_{inv}$ is equivalent to a sequence of inferences by ($T\vee$) and (C) in $\mathbf{C}_{inv}$. We can also see that the rules ($T\vee$) and (C) in $\mathbf{C}^{\xi}_{inv}$ have the same form as ($T\vee_1$) and (C) in $\mathbf{I}^{\xi}_{inv}$, respectively. Because the Instance Lemma is valid for $\mathbf{C}^{\xi}_{inv}$, it follows that we have to prove only that a sequence of two inferences by

In the rule (Ax) below, A and B are weakly unifiable atoms, ρ renames away A from B and τ is a most general unifier of $A\rho$ and B. In all rules the premises are variable-disjoint. In all rules each premise is also variable-disjoint with $var(\xi)$. In all rules θ is a most general unifier of σ_1, σ_2. In the rule $(T\vee_2)$ ϑ is a simultaneous most general unifier of the pairs σ_1, δ_1 and σ_2, δ_2. The rules $(\mathrm{F}\,\forall)$ and $(\mathrm{T}\,\exists)$ satisfy the eigenvariable condition: the term $x\sigma$ is a variable not occurring in the conclusion of the rule.

$$\frac{}{\mathrm{T}\,A\cdot\rho\tau, \mathrm{F}\,B\cdot\tau}\ (Ax)$$

$$\frac{\Gamma,\ \mathrm{T}\,A\cdot\sigma_1,\ \mathrm{T}\,A\cdot\sigma_2}{\Gamma\theta,\ \mathrm{T}\,A\cdot\sigma_1\theta}\ (C)$$

$$\frac{\Gamma,\ \mathrm{T}\,A\cdot\sigma}{\Gamma,\ \mathrm{T}\,A\wedge B\cdot\sigma}\ (\mathrm{T}\wedge_l) \qquad \frac{\Gamma,\ \mathrm{T}\,B\cdot\sigma}{\Gamma,\ \mathrm{T}\,A\wedge B\cdot\sigma}\ (\mathrm{T}\wedge_r)$$

$$\frac{\Gamma,\ \mathrm{F}\,A\cdot\sigma_1 \quad \Delta,\ \mathrm{F}\,B\cdot\sigma_2}{\Gamma\theta,\ \Delta\theta,\ \mathrm{F}\,A\wedge B\cdot\sigma_1\theta}\ (\mathrm{F}\wedge)$$

$$\frac{\Gamma,\ \mathrm{T}\,A\cdot\sigma_1 \quad \Delta,\ \mathrm{T}\,B\cdot\sigma_2}{\Gamma\theta,\ \Delta\theta,\ \mathrm{T}\,A\vee B\cdot\sigma_1\theta}\ (\mathrm{T}\vee_1)$$

$$\frac{\Gamma,\ \mathrm{T}\,A\cdot\sigma_1,\ \mathrm{F}\,C\cdot\delta_1 \quad \Delta,\ \mathrm{T}\,B\cdot\sigma_2,\ \mathrm{F}\,C\cdot\delta_2}{\Gamma\vartheta,\ \Delta\vartheta,\ \mathrm{T}\,A\vee B\cdot\sigma_1\vartheta,\ \mathrm{F}\,C\cdot\delta_1\vartheta}\ (\mathrm{T}\vee_2)$$

$$\frac{\Gamma,\ \mathrm{F}\,A\cdot\sigma}{\Gamma,\ \mathrm{F}\,A\vee B\cdot\sigma}\ (\mathrm{F}\vee_l) \qquad \frac{\Gamma,\ \mathrm{F}\,B\cdot\sigma}{\Gamma,\ \mathrm{F}\,A\vee B\cdot\sigma}\ (\mathrm{F}\vee_r)$$

$$\frac{\Gamma,\ \mathrm{F}\,A\cdot\sigma_1 \quad \Delta,\ \mathrm{T}\,B\cdot\sigma_2}{\Gamma\theta,\ \Delta\theta,\ \mathrm{T}\,A\supset B\cdot\sigma_1\theta}\ (\mathrm{T}\supset)$$

$$\frac{\Gamma,\ \mathrm{T}\,A\cdot\sigma}{\Gamma,\ \mathrm{F}\,A\supset B\cdot\sigma}\ (\mathrm{F}\supset_l) \qquad \frac{\Gamma,\ \mathrm{F}\,B\cdot\sigma}{\Gamma,\ \mathrm{F}\,A\supset B\cdot\sigma}\ (\mathrm{F}\supset_r)$$

$$\frac{\Gamma,\ \mathrm{T}\,A\cdot\sigma_1,\ \mathrm{F}\,A\supset B\cdot\sigma_2}{\Gamma\theta,\ \mathrm{F}\,A\supset B\cdot\sigma_2\theta}\ (\mathrm{F}\supset_{lr})$$

$$\frac{\Gamma,\ \mathrm{F}\,A\cdot\sigma}{\Gamma,\ \mathrm{T}\,\neg A\cdot\sigma}\ (\mathrm{T}\neg) \qquad \frac{\Gamma,\ \mathrm{T}\,A\cdot\sigma}{\Gamma,\ \mathrm{F}\,\neg A\cdot\sigma}\ (\mathrm{F}\neg)$$

$$\frac{\Gamma,\ \mathrm{T}\,A\cdot\sigma}{\Gamma,\ \mathrm{T}\,\forall xA\cdot\sigma_{-x}}\ (\mathrm{T}\forall) \qquad \frac{\Gamma,\ \mathrm{F}\,A\cdot\sigma}{\Gamma,\ \mathrm{F}\,\forall xA\cdot\sigma_{-x}}\ (\mathrm{F}\forall)$$

$$\frac{\Gamma,\ \mathrm{T}\,A\cdot\sigma}{\Gamma,\ \mathrm{T}\,\exists xA\cdot\sigma_{-x}}\ (\mathrm{T}\exists) \qquad \frac{\Gamma,\ \mathrm{F}\,A\cdot\sigma}{\Gamma,\ \mathrm{F}\,\exists xA\cdot\sigma_{-x}}\ (\mathrm{F}\exists)$$

Figure 17: Calculus $\mathbf{I}^{\xi}_{inv}$

$(T\vee)$ and (C) in $\mathbf{C}^{\xi}_{inv}$ is equivalent to an inference by $T\vee_2$ in $\mathbf{I}^{\xi}_{inv}$. In other words, the last sequents of derivations

$$\cfrac{\cfrac{\Gamma,\ \mathrm{T}\,A\cdot\sigma_1,\ \mathrm{F}\,C\cdot\delta_1 \quad \Delta,\ \mathrm{T}\,B\cdot\sigma_2,\ \mathrm{F}\,C\cdot\delta_2}{\Gamma\theta,\ \Delta\theta,\ \mathrm{T}\,A\vee B\cdot\sigma_1\theta,\ \mathrm{F}\,C\cdot\delta_1\theta,\ \mathrm{F}\,C\cdot\delta_2\theta}\,(\mathrm{T}\vee)}{\Gamma\theta\tau,\ \Delta\theta\tau,\ \mathrm{T}\,A\vee B\cdot\sigma_1\theta\tau,\ \mathrm{F}\,C\cdot\delta_1\theta\tau}\,(C)$$

and

$$\frac{\Gamma,\ \mathrm{T}\,A\cdot\sigma_1,\ \mathrm{F}\,C\cdot\delta_1 \quad \Delta,\ \mathrm{T}\,B\cdot\sigma_2,\ \mathrm{F}\,C\cdot\delta_2}{\Gamma\vartheta,\ \Delta\vartheta,\ \mathrm{T}\,A\vee B\cdot\sigma_1\vartheta,\ \mathrm{F}\,C\cdot\delta_1\vartheta}\,(\mathrm{T}\vee_2)$$

are the same. Using the standard arguments of unification theory[3] and the fact that for every variable z, if $z \notin (dom(\delta_1) \cup dom(\delta_2))$ then $z\delta_1\theta = z\delta_2\theta$, we prove that $\vartheta = \theta\tau$. □

This lemma and the arguments used to prove the completeness of $\mathbf{C}^{\xi}_{inv}$ yield the completeness theorem for $\mathbf{I}^{\xi}_{inv}$:

4.3. THEOREM (Completeness of $\mathbf{I}^{\xi}_{inv}$). *Let ξ be a closed signed formula. Then ξ is unsatisfiable in intuitionistic logic if and only if ξ has a refutation in $\mathbf{I}^{\xi}_{inv}$.* □

Consider an example refutation by the inverse method of the sequent

$$\mathrm{T}\,A\vee B,\ \mathrm{T}\,\forall x(A\supset P(x,b)),\ \mathrm{T}\,\forall x(B\supset P(a,x)),\ \mathrm{F}\,\exists x\exists y P(x,y). \tag{4.2}$$

We show the $\mathbf{I}^{\xi}_{inv}$-derivation of this sequent in Figure 18. This example demonstrates in particular the necessity of the special form for disjunction rule $(T\vee_2)$ in $\mathbf{I}^{\xi}_{inv}$ requiring simultaneous unification of two pairs of substitutions[4].

We would like to note that the resulting inverse method calculus varies depending on the initial tableau calculus. Had we started with the original calculus of Gentzen [1934], we would have obtained a different version of $\mathbf{I}^{\xi}_{inv}$. An essentially different calculus can be obtained from the multi-succedent sequent formalization of intuitionistic logic [Maehara 1954, Beth 1956], see also [Curry 1963, Troelstra and Schwichtenberg 1996], in which multiple signed formulas $\mathrm{F}\,A$ are allowed in sequents, but some rules are different. An interesting line of research is to obtain "more efficient" (from the viewpoint of the inverse method) calculi. The criteria for efficiency can be the length of propositional derivations, the smaller number of non-invertible rules, the smaller number of pairs of non-permutable rules etc. For tableau-based methods, this line of research is pursued in several publications,

[3] See e.g., [Apt 1990] where an equivalent notion of unification of a set of equations $\{s_1 = t_1, \ldots, s_n = t_n\}$ is considered instead of unification of tuples $\langle s_1, \ldots, s_n\rangle$ and $\langle t_1, \ldots, t_n\rangle$.

[4] [Tammet 1996, Tammet 1997] included a simplified version of the disjunction rule in his intuitionistic calculi for the inverse method. Essentially, he overlooked the necessity of unifying *two pairs* of substitutions simultaneously.

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{\mathrm{T}\,A,\ \mathrm{F}\,A \qquad \mathrm{T}\,P(x,b)\cdot\{x\mapsto x_1\},\ \mathrm{F}\,P(x,y)\cdot\{x\mapsto x_1, y\mapsto b\}}{\mathrm{T}\,A,\ \mathrm{T}\,A\supset P(x,b)\cdot\{x\mapsto x_1\},\ \mathrm{F}\,P(x,y)\cdot\{x\mapsto x_1, y\mapsto b\}}\,(\mathrm{T}\ \supset)
}{\mathrm{T}\,A,\ \mathrm{T}\,\forall x(A\supset P(x,b)),\ \mathrm{F}\,P(x,y)\cdot\{x\mapsto x_1, y\mapsto b\}}\,(\mathrm{T}\,\forall)
\quad
\dfrac{
\dfrac{\mathrm{T}\,B,\ \mathrm{F}\,B \qquad \begin{array}{l}\mathrm{T}\,P(a,x)\cdot\{x\mapsto x_2\},\\ \mathrm{F}\,P(x,y)\cdot\{x\mapsto a, y\mapsto x_2\}\end{array}}{\begin{array}{l}\mathrm{T}\,B,\ \mathrm{T}\,(B\supset P(a,x))\cdot\{x\mapsto x_2\},\\ \mathrm{F}\,P(x,y)\cdot\{x\mapsto a, y\mapsto x_2\}\end{array}}\,(\mathrm{T}\ \supset)
}{\mathrm{T}\,B,\ \mathrm{T}\,\forall x(B\supset P(a,x)),\ \mathrm{F}\,P(x,y)\cdot\{x\mapsto a, y\mapsto x_2\}}\,(\mathrm{T}\,\forall)
}{\mathrm{T}\,A\vee B,\ \mathrm{T}\,\forall x(A\supset P(x,b)),\ \mathrm{T}\,\forall x(B\supset P(a,x)),\ \mathrm{F}\,P(x,y)\cdot\{x\mapsto a, y\mapsto b\}}\,(\mathrm{T}\vee_2)
}{\mathrm{T}\,A\vee B,\ \mathrm{T}\,\forall x(A\supset P(x,b)),\ \mathrm{T}\,\forall x(B\supset P(a,x)),\ \mathrm{F}\,\exists y P(x,y)}\,(\mathrm{F}\,\exists)
}{\mathrm{T}\,A\vee B,\ \mathrm{T}\,\forall x(A\supset P(x,b)),\ \mathrm{T}\,\forall x(B\supset P(a,x)),\ \mathrm{F}\,\exists x\exists y P(x,y)}\,(\mathrm{F}\,\exists)
$$

Figure 18: A $\mathbf{I}^{\xi}_{inv}$-derivation

including [Dyckhoff 1992, Shankar 1992, Hudelmaier 1993], but the technique developed for tableau-based methods does not necessarily apply to the inverse method. For example, the calculus of Hudelmaier [1993] is extremely efficient for the tableau-based proof-search in intuitionistic propositional logic, but does not have the subformula property.

4.2. Modal logics

In this section, we cook the inverse method calculi for the propositional versions of several modal logics. As in the previous sections, we develop no semantics for these logics, instead we exploit proof-theoretic properties of cut-free sequent calculi for these logics. For standard formalizations and semantics of modal logics, see e.g. [Kripke 1963, Chagrov and Zakharyaschev 1996, Fitting and Mendelsohn 1998] and [Waaler 2001] (Chapter 22 of this Handbook). Proof-theoretic investigations of these logics can be found in [Fitting 1983]. The predicate versions of these logics are described in e.g. [Bowen 1979, Fitting 1983, Fitting and Mendelsohn 1998, Fitting, Thalmann and Voronkov 2000]. We discuss in some detail the design of the inverse method calculi for **K**, for other modal logics we simply give a table of inference rules, leaving the (trivial) proofs to the reader.

Formulas of modal logics have two additional unary connectives $\Box$ and $\Diamond$. For simplicity, we will only consider formulas in negation normal form, i.e. those built from literals using $\wedge$, $\vee$, $\forall$, $\exists$, $\Box$ and $\Diamond$. For a multiset of formulas Γ we denote by $\Box\Gamma$ the multiset of formulas consisting of formulas $\Box A$ for all $A \in \Gamma$. In order to obtain the negation norm form of a modal formula, we use the transformation rules given on page 204 plus the following rules:

$$\begin{array}{rcl} \neg\Box A & \Rightarrow & \Diamond\neg A, \\ \neg\Diamond A & \Rightarrow & \Box\neg A. \end{array}$$

As in the case of negation normal forms for classical logic, the sequents are multisets of formulas (not signed formulas).

4.2.1. Modal logic **K**

The sequent calculus $\mathbf{K}_{tab}$ is obtained from the sequent calculus $\mathbf{C}_{tab}$ for NNFs (Figure 10 on page 205) by adding one inference rule

$$\frac{\Delta, A}{\Gamma, \Box\Delta, \Diamond A}\ (\Diamond). \tag{4.3}$$

An example refutation in $\mathbf{K}_{tab}$ of a propositional modal formula is shown below.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{P, \neg P, \neg Q \quad P, Q, \neg Q}{P, \neg P \vee Q, \neg Q}\ (\vee)}{\Box P, \Box(\neg P \vee Q), \Diamond\neg Q, \Diamond Q}\ (\Diamond)}{\Box P \wedge \Box(\neg P \vee Q), \Diamond\neg Q, \Diamond Q}\ (\wedge)}{\Box P \wedge \Box(\neg P \vee Q) \wedge \Diamond\neg Q, \Diamond Q}\ (\wedge)}{\Box P \wedge \Box(\neg P \vee Q) \wedge \Diamond\neg Q \wedge \Diamond Q}\ (\wedge). \tag{4.4}$$

As we can see, this logic also has the subformula property.

We follow the universal recipe of automated deduction. At our disposal are the initial ingredients: the rules of $\mathbf{K}_{tab}$. Our next aim is to design a suitable calculus dealing with closed formulas (the ground version).

Since we are now much more experienced with cooking, we know an easy way to cook: think of the Subsequent Lemma. In order to find a calculus suitable for the inverse method, we have to extend $\mathbf{C}^G_{inv}$ by rules corresponding to $(\Diamond)$ so that the Subsequent Lemma remains true. We will illustrate how such rules can be obtained in an essentially mechanical way by trying to adapt the techniques used in the proofs of this key lemma.

For the Subsequent Lemma to remain true we have to find an inference rule (or a collection of inference rules) with the following two properties: first, the rule should be sound for $\mathbf{K}$ and second, given a derivation of a subsequent of Δ, A we should be able to derive a subsequent of $\Gamma, \Box\Delta, \Diamond A$. Let us try the most obvious candidate:

$$\frac{\Delta, A}{\Box\Delta, \Diamond A}\ (\Diamond). \tag{4.5}$$

Obviously, this rule is sound because it is a special case of (4.3) with $\Gamma = \emptyset$.

Let us verify the second property. Suppose that a subsequent of Δ, A is derivable. Consider two cases.

1. This subsequent contains A, then it has the form Δ', A for $\Delta' \dot{\subseteq} \Delta$. We can apply an instance of rule (4.5):

$$\frac{\Delta', A}{\Box\Delta', \Diamond A}\ (\Diamond)$$

to obtain a subsequent of $\Gamma, \Box\Delta, \Diamond A$.

2. The subsequent of Δ, A does not contain A. In this case, it has the form Δ' for $\Delta' \dot{\subseteq} \Delta$. It seems that in general there is no way to derive a subsequent of $\Gamma, \Box\Delta, \Diamond A$ from Δ' only. Therefore, we add one more rule to the inverse method calculus:

$$\frac{\Delta}{\Box\Delta, \Diamond A}\ (\Diamond^+). \tag{4.6}$$

We leave the proof of soundness of this rule to the reader. With the new rule, the Subsequent Lemma holds, since now we can use the inference

$$\frac{\Delta'}{\Box\Delta', \Diamond A}\ (\Diamond^+)$$

to obtain a subsequent of $\Gamma, \Box\Delta, \Diamond A$.

The new rules create no troubles for the Instance Lemma.

By adding to $\mathbf{C}_{inv}$ rules $(\Diamond)$ and $(\Diamond^+)$ we obtain the ***calculus*** $\mathbf{K}_{inv}$.

Our next aim is to add unification to $\mathbf{K}_{inv}$ and cook this calculus into a free-variable calculus. With our greater experience in cooking, this is almost mechanical. Let G be a formula. By adding to $\mathbf{C}^G_{inv}$ the free-variable versions of these rules

$$\frac{\Delta, A \cdot \sigma}{\Box\Delta, \Diamond A \cdot \sigma}\ (\Diamond) \qquad \text{and} \qquad \frac{\Delta}{\Box\Delta, \Diamond A \cdot \varepsilon}\ (\Diamond^{+})$$

we obtain the ***calculus*** $\mathbf{K}^{G}_{inv}$. The proof of the Instance Lemma for $\mathbf{K}^{G}_{inv}$ is also straightforward.

Below we give an example derivation in $\mathbf{K}_{inv}$ which can be obtained by applying the transformation from the Subsequent Lemma to refutation (4.4)

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{P, \neg P \quad Q, \neg Q}{P, \neg P \vee Q, \neg Q}\ (\vee)}{\Box P, \Box(\neg P \vee Q), \Diamond\neg Q}\ (\Diamond)}{\begin{array}{c}\Box P,\ \Box(\neg P \vee Q) \wedge \Diamond\neg Q \wedge \Diamond Q, \\ \Box(\neg P \vee Q) \wedge \Diamond\neg Q \wedge \Diamond Q\end{array}}\ (\wedge)}{\Box P, \Box(\neg P \vee Q) \wedge \Diamond\neg Q \wedge \Diamond Q}\ (C)}{\Box P \wedge \Box(\neg P \vee Q) \wedge \Diamond\neg Q \wedge \Diamond Q, \Box P \wedge \Box(\neg P \vee Q) \wedge \Diamond\neg Q \wedge \Diamond Q}\ (\wedge)}{\Box P \wedge \Box(\neg P \vee Q) \wedge \Diamond\neg Q \wedge \Diamond Q}\ (C)$$

4.3. Other modal logics

In this section we give the inverse method calculi for propositional modal logics **T**, **D**, **S4**, **K4**, and **D4**. Cut-free calculi for these logics used here are taken from [Fitting 1983], but presented in a different notation. Instead of giving a detailed presentation, we simply give a table of inference rules for these logics, both in the tableau form and the inverse form. These rules are given in Figure 19.

The reader can check the subsequent lemma for these calculi. Note that the second $(\Diamond)$ rules of the inverse calculi for **D** and **D4** obtained by a direct translation of the $(\Diamond)$ rules of the tableau calculi, are redundant. Indeed, they can be replaced by the respective $(\Box)$ rules.

5. Naming and connections with resolution

Formulas are too complex objects to be manipulated. During proof-search by the inverse method, sequents tend to grow and become difficult to manage. In this section we consider the *naming technique* that allows one to have a convenient representation of formulas by atoms. The naming technique was already used by Maslov [1968]. The naming technique is also used in other methods of theorem proving in nonclassical logics (e.g., [Horrocks and Patel-Schneider 1999, de Nivelle, Schmidt and Hustadt 2000]), though usually it is not formalized as a logical calculus. Voronkov [2001] introduces a so-called *path calculus* where paths are used instead of the names. The use of the paths considerably simplifies the completeness proof for some redundancies. We consider path calculi in Section 7 of this chapter.

logic	tableau rules	inverse rules
T	$\dfrac{\Gamma, A}{\Box\Gamma, \Diamond A, \Delta}\ (\Diamond)$ $\dfrac{\Gamma, A}{\Gamma, \Box A}\ (\Box)$	$\dfrac{\Gamma, A}{\Box\Gamma, \Diamond A}\ (\Diamond)$ $\quad \dfrac{\Gamma}{\Box\Gamma, \Diamond A}\ (\Diamond)$ $\dfrac{\Gamma, A}{\Gamma, \Box A}\ (\Box)$
D	$\dfrac{\Gamma, A}{\Box\Gamma, \Diamond A, \Delta}\ (\Diamond)$ $\dfrac{\Gamma}{\Box\Gamma, \Delta}\ (\Box)$	$\dfrac{\Gamma, A}{\Box\Gamma, \Diamond A}\ (\Diamond)$ $\quad \dfrac{\Gamma}{\Box\Gamma, \Diamond A}\ (\Diamond)$ $\dfrac{\Gamma}{\Box\Gamma}\ (\Box)$
S4	$\dfrac{\Box\Gamma, A}{\Box\Gamma, \Diamond A, \Delta}\ (\Diamond)$ $\dfrac{\Gamma, A}{\Gamma, \Box A}\ (\Box)$	$\dfrac{\Box\Gamma, A}{\Box\Gamma, \Diamond A}\ (\Diamond)$ $\dfrac{\Gamma, A}{\Gamma, \Box A}\ (\Box)$
D4	$\dfrac{\Gamma, \Box\Gamma, A}{\Box\Gamma, \Diamond A, \Delta}\ (\Diamond)$ $\dfrac{\Gamma, \Box\Gamma}{\Box\Gamma, \Delta}\ (\Box)$	$\dfrac{\Gamma_1, \Box\Gamma_2, A}{\Box\Gamma_1, \Box\Gamma_2, \Diamond A}\ (\Diamond)$ $\quad \dfrac{\Gamma_1, \Box\Gamma_2}{\Box\Gamma_1, \Box\Gamma_2, \Diamond A}\ (\Diamond)$ $\dfrac{\Gamma_1, \Box\Gamma_2}{\Box\Gamma_1, \Box\Gamma_2}\ (\Box)$
K4	$\dfrac{\Gamma, \Box\Gamma, A}{\Box\Gamma, \Diamond A, \Delta}\ (\Diamond)$	$\dfrac{\Gamma_1, \Box\Gamma_2, A}{\Box\Gamma_1, \Box\Gamma_2, \Diamond A}\ (\Diamond)$ $\quad \dfrac{\Gamma_1, \Box\Gamma_2}{\Box\Gamma_1, \Box\Gamma_2, \Diamond A}\ (\Diamond)$

The second $(\Diamond)$ rules of the inverse calculi for **D** and **D4** are redundant.

Figure 19: Inference rules for several modal logics

$\exists y \forall x \forall z (P(z,x) \wedge (\neg P(z,fx) \vee \neg P(x,ffy)))$	G	$\exists y A(y)$
$\forall x \forall z (P(z,x) \wedge (\neg P(z,fx) \vee \neg P(x,ffy)))$	$A(y)$	$\forall x B(x,y)$
$\forall z (P(z,x) \wedge (\neg P(z,fx) \vee \neg P(x,ffy)))$	$B(x,y)$	$\forall z C(z,x,y)$
$P(z,x) \wedge (\neg P(z,fx) \vee \neg P(x,ffy))$	$C(z,x,y)$	$D(z,x) \wedge E(z,x,y)$
$P(z,x)$	$D(z,x)$	
$\neg P(z,fx) \vee \neg P(x,ffy)$	$E(z,x,y)$	$H(z,x) \vee I(x,y)$
$\neg P(z,fx)$	$H(z,x)$	
$\neg P(x,ffy)$	$I(x,y)$	

Table 1: Subformulas and names

5.1. Naming

In this section we will restrict ourselves to formulas in negation normal form and consider formulas instead of signed formulas. We will try to avoid complete formalization, presenting the technique by examples.

The essence of the naming technique is the following: with each free subformula A of the goal formula G, we associate a unique predicate symbol N, and use these predicate symbols to name arbitrary subformulas in the following way. Let $A(x_1,\dots,x_n)$ be a free subformula of G, where $x_1,\dots,x_n$ are all free variables of this subformula and N be the predicate symbol associated with $A(x_1,\dots,x_n)$, then any subformula $A(t_1,\dots,t_n)$ of G will be represented by the atomic formula $N(t_1,\dots,t_n)$. We say that the atomic formula $N(t_1,\dots,t_n)$ is a ***name*** of the subformula $A(t_1,\dots,t_n)$.

Since every inference rule of $\mathbf{C}^G_{inv}$ is defined by its principal formula, that is a subformula of G, we also obtain a convenient representation of inference rules. To illustrate the technique, we consider formula (3.7) in negation normal form. The free subformulas of this formula are shown in the left-hand column of Table 1.

We associate with these free subformulas new predicate symbols $A,\dots,H$, then the names of these subformulas are shown in the middle column of Table 1. In the right-hand column of the table we show the representation of the subformulas of G through the names of its immediate subformulas.

Formally, let G be the goal formula, $N_1, N_2, \dots$ be an alphabet of predicate symbols not occurring in G and $x_1, x_2, \dots$ be all variables of G. Let all free subformulas of G be $A_1,\dots,A_n$. Let $\bar{y}_1,\dots,\bar{y}_n$ be the sequences of variables of $A_1,\dots,A_n$, respectively, the variables in each sequence $\bar{y}_i$ are in the order of their first occurrences in A_i. We call the atomic formulas $N_i(\bar{y}_i)$ the ***names*** of the free subformulas $A_i(\bar{y}_i)$ of G. Likewise, let $A_i(\bar{t})$ be any subformula of G. We call the atomic formula $N_i(\bar{t})$ the *name* of $A_i(\bar{t})$.

For example, in Table 1 the atomic formula $E(z,x,y)$ is the name of the free subformula $\neg P(z,fx) \vee \neg P(x,ffy)$ of G. Then for every term t_z, t_x, t_y, the atomic formula $E(t_z, t_x, t_y)$ is the name of the subformula $\neg P(t_z, ft_x) \vee \neg P(t_x, fft_y)$ of G.

We will now modify the calculus $\mathbf{C}^G_{inv}$ by replacing subformulas by their names. We call the resulting calculus the ***name calculus for*** G. Every non-literal free subformula A of G induces one or more inference rules of $\mathbf{C}^G_{inv}$ with A as the principal formula. Consider, for example, the free subformula $\neg P(z, fx) \vee \neg P(x, ffy)$. The calculus $\mathbf{C}^G_{inv}$ contains the following inference rule

$$\frac{\Gamma,\ \neg P(z, fx) \cdot \sigma_1 \quad \Delta,\ \neg P(x, ffy) \cdot \sigma_2}{\Gamma\theta,\ \Delta\theta,\ \neg P(z, fx) \vee \neg P(x, ffy) \cdot \sigma_1\theta}\ (\vee). \tag{5.1}$$

Note that σ_1 has the form $\{z \mapsto s_z, x \mapsto s_x\}$, for some terms s_z, s_x. Likewise, σ_2 has the form $\{x \mapsto t_x, y \mapsto t_y\}$, for some terms t_x, t_y. It is not hard to argue that θ has the form $\{z \mapsto s_z, y \mapsto t_y\}\tau$, where τ is a most general unifier of s_x and t_x. Therefore, the rule can be rewritten as

$$\frac{\Gamma,\ \neg P(s_z, fs_x) \quad \Delta,\ \neg P(t_x, fft_y)}{(\Gamma,\ \Delta,\ (\neg P(s_z, fs_x) \vee \neg P(s_x, fft_y))\tau}\ (\vee), \tag{5.2}$$

where τ is a most general unifier of s_x, t_x. By replacing the principal and side formulas of this rule by their names, we obtain the following inference rule of the name calculus for G:

$$\frac{\Gamma,\ H(s_z, s_x) \quad \Delta,\ I(t_x, t_y)}{(\Gamma,\ \Delta,\ E(s_z, s_x, t_y))\tau}\ (\vee), \tag{5.3}$$

where τ is a most general unifier of s_x and t_x.

Consider another example. Since the goal formula G contains occurrences of two literals $P(z, x)$, $\neg P(z, fx)$, the calculus $\mathbf{C}^G_{inv}$ contains the axiom

$$P(z, x) \cdot \{x \mapsto fx_2, z \mapsto x_1\},\ \neg P(z, fx) \cdot \{x \mapsto x_2, z \mapsto x_1\}.$$

By replacing the two literals by their names and applying the substitutions, we obtain the following axiom of the name calculus for G:

$$D(x_1, fx_2),\ H(x_1, x_2).$$

In total, the name calculus for $\exists y \forall x \forall z (P(z, x) \wedge (\neg P(z, fx) \vee \neg P(x, ffy)))$ contains the axioms and inference rules shown in Figure 20 (compare them with the subformulas and names of Table 1):

An example derivation in the name calculus is given in Figure 21. This derivation can be obtained from the derivation of Figure 14 by replacing subformulas by their names. Note that the derivation in the name calculus is much more compact than the original derivation in $\mathbf{C}^G_{inv}$.

5.2. *The inverse method as resolution*

Sequents in name calculi are multisets of atoms, i.e., they can be considered as (positive) clauses in resolution. In this paper a ***clause*** is a multiset of literals.

$$\frac{\Gamma,\ A(s_y)}{\Gamma,\ G}\ (\exists) \qquad \frac{\Gamma,\ B(s_x,s_y)}{\Gamma,\ A(s_y)}\ (\forall_1) \qquad \frac{\Gamma,\ C(s_z,s_x,s_y)}{\Gamma,\ B(s_x,s_y)}\ (\forall_2)$$

$$\frac{\Gamma,\ D(s_z,s_x)}{\Gamma,\ C(s_z,s_x,y)}\ (\wedge_l) \qquad \frac{\Gamma,\ E(s_z,s_x,s_y)}{\Gamma,\ C(s_z,s_x,s_y)}\ (\wedge_r)$$

$$\frac{\Gamma,\ H(s_z,s_x) \quad \Delta,\ I(t_x,t_y)}{(\Gamma,\ \Delta,\ E(s_z,s_x,t_y))\mathrm{mgu}(s_x,t_x)}\ (\vee)$$

$$\frac{}{D(x_1,fx_2),\ H(x_1,x_2).}\ (Ax_1) \qquad \frac{}{D(x_1,ffx_2),\ I(x_1,x_2).}\ (Ax_2)$$

$$\frac{\Gamma,F_1,F_2}{(\Gamma,F_1)\sigma}\ (C)$$

The rule $(\exists)$ satisfies the eigenvariable condition: s_y is a variable not occurring in Γ. In the contraction rule (C) σ a most general unifier of F_1 and F_2.

Figure 20: The name calculus for $\exists y\forall x\forall z(P(z,x)\wedge(\neg P(z,fx)\vee\neg P(x,ffy)))$

(1)	$D(x_1,fx_2),\ H(x_1,x_2)$	(Ax_1)
(2)	$D(y_1,ffy_2),\ I(y_1,y_2)$	(Ax_2)
(3)	$D(x_1,fx_2),\ D(x_2,ffx_3),\ E(x_1,x_2,x_3)$	$(\vee)$ from (1,2)
(4)	$D(fx_1,ffx_1),\ E(fx_1,fx_1,x_1)$	(C) from (3)
(5)	$D(fx_1,ffx_1),\ C(fx_1,fx_1,x_1)$	$(\wedge_r)$ from (4)
(6)	$D(fx_1,ffx_1),\ B(fx_1,x_1)$	$(\forall_2)$ from (5)
(7)	$D(fx_1,ffx_1),\ A(x_1)$	$(\forall_1)$ from (6)
(8)	$C(fx_1,ffx_1,x_2),\ A(x_1)$	$(\wedge_l)$ from (7)
(9)	$B(ffx_1,x_2),\ A(x_1)$	$(\forall_2)$ from (8)
(10)	$A(x_2),\ A(x_1)$	$(\forall_1)$ from (9)
(11)	$A(x_1)$	(C) from (10)
(12)	G	$(\exists)$ from (11)

Figure 21: Derivation in the name calculus corresponding to the derivation of Figure 14.

There exists a close relationship between inferences in the name calculus and some inferences by hyperresolution. Consider, for example, inference rule (5.3) of the name calculus. One possible inference by this rule (taken from the derivation of Figure 21) is

$$\frac{D(x_1, fx_2),\ H(x_1, x_2) \quad D(x_4, ffx_3),\ I(x_4, x_3)}{D(x_1, fx_2),\ D(x_2, ffx_3),\ E(x_1, x_2, x_3)}\ (\vee).$$

Note that exactly the same conclusion can be obtained by the following application of the positive hyperresolution rule[5] with the additional clause $\neg H(z, x),\ \neg I(x, y),\ E(z, x, y)$.

$$\frac{D(x_1, fx_2),\ H(x_1, x_2) \quad D(x_4, ffx_3),\ I(x_4, x_3) \quad \neg H(z, x),\ \neg I(x, y),\ E(z, x, y)}{D(x_1, fx_2),\ D(x_2, ffx_3),\ E(x_1, x_2, x_3)}\ (hyper).$$

This observation can be generalized: every inference by (5.3) in the name calculus can be made into an inference by hyperresolution by adding the additional premise $\neg H(z, x),\ \neg I(x, y),\ E(z, x, y)$. Every such inference by hyperresolution with one of the premises $\neg H(z, x),\ \neg I(x, y),\ E(z, x, y)$ can be made into an inference by (5.3) by removing this premise.

Likewise, any rule $(\vee)$ of the name calculus for any formula G can be simulated by hyperresolution. Moreover, this is true for every rule $(\wedge_l)$, $(\wedge_r)$ and $(\forall)$. The only rule that cannot be directly simulated by hyperresolution is $(\exists)$, because of the eigenvariable condition. However, the quantifier $\exists$ never occurs in Skolemized formulas, so for Skolemized formulas the inverse method can be simulated by hyperresolution.

Formally, the transformation works as follows. Let G be a rectified formula in negation normal form without occurrences of $\exists$. We will now build two sets of clauses: a set Ax_G of positive clauses and a set $Rules_G$ of non-positive clauses. ***The set*** Ax_G is exactly the set of axioms of the name calculus for G and can be described as follows.

- Suppose that A, B is a pair of atomic formulas such that A and $\neg B$ are free subformulas of G, and A and B are weakly unifiable. Then Ax_G contains a clause $A'\rho\tau, B'\tau$ such that A', B' are the names of $A, \neg B$ respectively, ρ renames A away from B, and τ is a most general unifier of $A\rho$ and B.

The ***set*** $Rules_G$ is described as follows.

- Suppose that $A \vee B$ is a free subformula of G, and the atoms A', B', C' are the names of $A, B, A \vee B$, respectively. Then $Rules_G$ contains the clause $\neg A', \neg B', C'$.
- Suppose that $A \wedge B$ is a free subformula of G, and the atoms A', B', C' are the names of $A, B, A \wedge B$, respectively. Then $Rules_G$ contains the clauses $\neg A', C'$ and $\neg B', C'$.

[5]Hyperresolution is discussed in [Bachmair and Ganzinger 2001, page 61] (Chapter 2 of this Handbook).

- Suppose that $\forall x A$ is a free subformula of G, and the atoms A', B' are the names of $A, \forall x A$, respectively. Then $Rules_G$ contains the clause $\neg A', B'$.

5.1. LEMMA (Bisimulation of Inferences). *(i) For every inference different from contraction*

$$\frac{A_1 \quad \cdots \quad A_n}{A}$$

in the name calculus for G, there exists a clause $C \in Rules_G$ such that

$$\frac{A_1 \quad \cdots \quad A_n \quad C}{A}$$

is a inference by positive hyperresolution.
(ii) For every clause $C \in Rules_G$ and every inference by positive hyperresolution

$$\frac{A_1 \quad \cdots \quad A_n \quad C}{A}$$

the figure

$$\frac{A_1 \quad \cdots \quad A_n}{A}$$

is a valid inference in the name calculus for G.

We leave the (straightforward) proof of this lemma to the reader.

Since the axioms in the name calculus for G are precisely the clauses in Ax_G, Lemma 5.1 implies the following lemma.

5.2. LEMMA (Bisimulation of Derivations). *(i) Every derivation $\mathcal{N}$ in the name calculus for G can be made into a positive hyperresolution derivation from the set of clauses $Ax_G \cup Rules_G$ by adding a clause from $Rules_G$ to the premises of inferences of $\mathcal{N}$ different from (Ax) or (C).*
(ii) For every derivation $\mathcal{D}$ by positive hyperresolution from the set of clauses $Ax_G \cup Rules_G$ the figure obtained from $\mathcal{D}$ by removing all clauses in $Rules_G$ is a valid derivation in the name calculus for G.

5.3. EXAMPLE. Let us illustrate this theorem. Consider the Skolemized form of formula (3.7) used in Example 3.21 on page 206:

$$G = \forall x \forall z (P(z, x) \wedge (\neg P(z, fx) \vee \neg P(x, ffc))).$$

Introduce the following names for the free subformulas of G:

$$\begin{array}{rcl}
H & = & \forall x \forall z (P(z, x) \wedge (\neg P(z, fx) \vee \neg P(x, ffc))), \\
A(x) & = & \forall z (P(z, x) \wedge (\neg P(z, fx) \vee \neg P(x, ffc))), \\
B(x, z) & = & P(z, x) \wedge (\neg P(z, fx) \vee \neg P(x, ffc)), \\
C(z, x) & = & P(z, x), \\
D(z, x) & = & \neg P(z, fx) \vee \neg P(x, ffc), \\
E(z, x) & = & \neg P(z, fx), \\
F(x) & = & \neg P(x, ffc).
\end{array}$$

The set of clauses Ax_G consists of two clauses:

$$\begin{array}{ll} (1) & C(z, fx),\ E(z, x). \\ (2) & C(z, ffc),\ F(z). \end{array}$$

The set of clauses $Rules_G$ consists of five clauses:

$$\begin{array}{ll} (3) & \neg A(x),\ H. \\ (4) & \neg B(x, z),\ A(x). \\ (5) & \neg C(z, x),\ B(x, z). \\ (6) & \neg D(z, x),\ B(x, z). \\ (7) & \neg E(z, x),\ \neg F(x),\ D(z, x). \end{array}$$

A hyperresolution derivation of H from the initial set of clauses (1)–(7) is as follows:

(8)	$B(fx, z),\ E(z, x)$	hyper from (1,5).
(9)	$B(ffc, z),\ F(z)$	hyper from (2,5).
(10)	$B(fx, z),\ B(ffc, x),\ D(z, x)$	hyper from (8,9,7).
(11)	$B(ffc, fc),\ B(ffc, fc),\ D(fc, fc)$	factor from (10).
(12)	$B(ffc, fc),\ D(fc, fc)$	factor from (10).
(13)	$B(ffc, fc),\ B(fc, fc)$	hyper from (12,6).
(14)	$A(ffc),\ B(fc, fc)$	hyper from (13,4).
(15)	$A(ffc),\ A(fc)$	hyper from (14,4).
(16)	$H,\ A(fc)$	hyper from (15,3).
(17)	$H,\ H$	hyper from (16,3).
(18)	H	factor from (17).

The reader can try to convert this derivation into a Gentzen-style derivation of the original formula, for example in the calculus $\mathbf{C}_{inv}$.

In resolution theorem proving, we are usually interested in finding a *refutation* from a set of clauses, instead of constructing a derivation of a goal. However, when the goal is a ground atom G, it is derivable from a set of clauses S if and only if the set of clauses $S \cup \{\neg G\}$ is inconsistent. Therefore we have

5.4. THEOREM. *Let G be a closed formula in negation normal form without occurrences of $\exists$ and H be the name of G. Then G is inconsistent if and only if so is the set of clauses $Ax_G \cup Rules_G \cup \{\neg H\}$.* □

The use of the naming technique and the simulation of the inverse method by resolution are interesting for several reasons. One reason is the simple representation of sequents by clauses. Another reason is that some strategies and redundancies known for resolution can be transferred from resolution to the inverse method. This is not so important for theorem proving in classical logic, where we can use the best implementations of resolution on the transformed set of clauses directly,[6] but this is invaluable for nonclassical logics. Indeed, as demonstrated in [Voronkov 1992], many strategies complete for resolution for classical logic have analogues complete for nonclassical logics.

5.3. Optimized translation

We will demonstrate how the use of some elementary properties of resolution-based theorem proving can improve the inverse method for classical logic, by formulating a more concise name calculus. In this name calculus only so-called *disjunctive subformulas* have names.

We will use the following simple property of clause form logic.

5.5. Lemma (Definition Elimination). *Suppose that P is a predicate symbol and S is a set of clauses. Suppose that P has at most one occurrence in each clause in S. Denote by S' the set of clauses obtained from S by adding, for each pair of clauses $P(\bar{s}) \wedge C$ and $\neg P(\bar{t}), D$ such that $\bar{s}$ and $\bar{t}$ are unifiable, the clause $(C, D)\mathrm{mgu}(\bar{s}, \bar{t})$ and by S'' the clause obtained from S' by deleting all clauses in which P occurs. Then S is inconsistent if and only if so is S''.* ☐

We will now show how to apply this lemma and Theorem 5.4 to obtain an optimized translation into a set of clauses. Using this optimized translation we can design a new calculus for Skolemized formulas of classical logic with no $(\wedge_l)$, $(\wedge_r)$, or $(\forall)$ rules, so the only remaining rules are $(\vee)$. We call ***disjunctive free subformulas of*** G all free subformulas F_1 and F_2 of G such that $F_1 \vee F_2$ is a free subformula of G. Our aim is now to eliminate names for all non-disjunctive free subformulas from the name calculus. This can be done by a straightforward application of the Definition Elimination Lemma.

5.6. Example. We will illustrate the optimized translation on the formula of Example 5.3 on page 225. Let us show how to eliminate all names for non-disjunctive subformulas. Before eliminating the names, let us first add to the set of clauses the negation of the name of the goal formula $\neg H$. Note that there are only two names of disjunctive free subformulas, namely $E(z, x)$ and $F(x)$. Elimination of all other names is illustrated in Figure 22. At each step of the elimination procedure we shade the literals with the currently eliminated atom, corresponding to the literals $P(\bar{s})$ and $\neg P(\bar{t})$ of the Definition Elimination Lemma.

It is not hard to argue that after a finite number of steps all names for non-disjunctive free subformulas will be eliminated. Moreover, the size of the resulting

[6] Indeed the currently fastest methods of first-order theorem proving take from the inverse method only the idea of the short clause form translation and then apply resolution strategies to the obtained "short" set of clauses [Weidenbach, Gaede and Rock 1996, Nonnengart and Weidenbach 1998].

$$
\begin{array}{rl}
(1) & C(z,fx),\ E(z,x).\\
(2) & C(z,ffc),\ F(z).\\
(3) & \neg A(x),\ H.\\
(4) & \neg B(x,z),\ A(x).\\
(5) & \neg C(z,x),\ B(x,z).\\
(6) & \neg D(z,x),\ B(x,z).\\
(7) & \neg E(z,x),\ \neg F(x),\ D(z,x).\\
(8) & \neg H.
\end{array}
\Rightarrow
\begin{array}{rl}
(1) & C(z,fx),\ E(z,x).\\
(2) & C(z,ffc),\ F(z).\\
(3) & \neg A(x),\ H.\\
(9) & \neg C(z,x),\ A(x).\\
(10) & \neg D(z,x),\ A(x).\\
(7) & \neg E(z,x),\ \neg F(x),\ D(z,x).\\
(8) & \neg H.
\end{array}
\Rightarrow
$$

$$
\begin{array}{rl}
(1) & C(z,fx),\ E(z,x).\\
(2) & C(z,ffc),\ F(z).\\
(3) & \neg A(x),\ H.\\
(9) & \neg C(z,x),\ A(x).\\
(11) & \neg E(z,x),\ \neg F(x),\ A(x).\\
(8) & \neg H.
\end{array}
\Rightarrow
\begin{array}{rl}
(1) & C(z,fx),\ E(z,x).\\
(2) & C(z,ffc),\ F(z).\\
(12) & \neg C(z,x),\ H.\\
(13) & \neg E(z,x),\ \neg F(x),\ H.\\
(8) & \neg H.
\end{array}
\Rightarrow
$$

$$
\begin{array}{rl}
(1) & C(z,fx),\ E(z,x).\\
(2) & C(z,ffc),\ F(z).\\
(13) & \neg C(z,x).\\
(14) & \neg E(z,x),\ \neg F(x).
\end{array}
\Rightarrow
\begin{array}{rl}
(15) & E(z,x).\\
(16) & F(z).\\
(14) & \neg E(z,x),\ \neg F(x).
\end{array}
$$

Figure 22: Elimination of names for non-disjunctive subformulas

set of clauses is less than the size of the original set. In fact, we can also eliminate some names for disjunctive subformulas as well, but this may result in the exponential blow-up of the size of the set of clauses.

We have demonstrated how to obtain the optimized translation by using definition elimination from the result of the nonoptimized translation, but the optimized translation can be formulated directly in terms of the goal formula. Essentially, instead of using the name of a subformula, one has to use the name of its *least disjunctive superformula*; see for details [Degtyarev and Voronkov 1994*b*, Degtyarev and Voronkov 1995*a*].

The technique that allows one to use only the names for disjunctive subformulas was used in a number of papers [Voronkov 1985, Voronkov 1990*b*, Degtyarev and Voronkov 1994*a*, Degtyarev and Voronkov 1995*a*, Degtyarev and Voronkov 1996*g*]. Similar technique leading to more concise name calculi for nonclassical logics were described in various forms in [Voronkov 1992, Mints et al. 1996, Voronkov 2001].

5.4. *Why hyperresolution*

Theorem 5.4 can be proved by simpler techniques as demonstrated in [Boy de la Tour 1990, Degtyarev and Voronkov 1994*a*]. The transformation of G into clausal form can be performed in polynomial time, unlike the standard transformation that can lead to exponentially large clausal forms. The use of the naming technique together with other techniques for obtaining short clausal forms is discussed in [Nonnengart and Weidenbach 2001] (Chapter 6 of this Handbook).

The use of hyperresolution on the transformed set of clauses is motivated by one observation. Since there is a *bisimulation* between inferences in the name calculus for G and hyperresolution inferences, the hyperresolution derivations can be transformed back to the derivations in the name calculus, which, in turn, have a simple translation into cut-free Gentzen-style derivations. Thus, we can easily transform hyperresolution derivations into Gentzen-style derivations.

Note that the result of a naming transformation is a set of clauses that is unsatisfiable if and only if so is the original formula. In general, it is not necessary to use positive hyperresolution in order to establish inconsistency of a set of clauses. Any other resolution-based methods can be used instead.

5.5. *Inference rules not captured by hyperresolution*

Italians believe that fish should never be cooked with cheese. This belief is not shared by some other cuisines. What happens when we deal with a calculus which has rules that do not fit into the (hyper)resolution framework, but still want to present them in the form of a resolution calculus? The result is something that is liked by some people and regarded as inappropriate cooking style by others.

A rich source of unusual inference rules is intuitionistic logic (both propositional and predicate). Let us consider several examples.

We will assume that we use a naming technique, so that each free-subformula of the goal signed formula is given a name, that is a predicate symbol, and try to simulate inference rules of the inverse method by special kind of clauses, like the clauses in $Rules_G$ used in the naming transformation for classical logic. We will denote the name of a free signed subformula α by N_α.

Most rules of $\mathbf{I}^{\xi}_{inv}$ of Figure 17 on page 214 can be easily presented as resolution rules. For example, the rule

$$\frac{\Gamma,\ \mathrm{F}\,A\cdot\sigma_1 \quad \Delta,\ \mathrm{F}\,B\cdot\sigma_2}{\Gamma\theta,\ \Delta\theta,\ \mathrm{F}\,A\wedge B\cdot\sigma_1\theta}\ (\mathrm{F}\wedge)$$

can be simulated by adding the clause

$$\neg N_{\mathrm{F}\,A}(\bar{x}), \neg N_{\mathrm{F}\,B}(\bar{y}), N_{\mathrm{F}\,A\wedge B}(\bar{z}),$$

where $\bar{x}, \bar{y}, \bar{z}$ are the all variables of the respective signed subformulas. Then any inference by $(\mathrm{F}\wedge)$ is simulated by two resolution inferences or one hyperresolution inference with this clause. However, the rule

$$\frac{\Gamma,\ \mathrm{T}\,A\cdot\sigma_1,\ \mathrm{F}\,C\cdot\delta_1 \quad \Delta,\ \mathrm{T}\,B\cdot\sigma_2,\ \mathrm{F}\,C\cdot\delta_2}{\Gamma\vartheta,\ \Delta\vartheta,\ \mathrm{T}\,A\vee B\cdot\sigma_1\vartheta,\ \mathrm{F}\,C\cdot\delta_1\vartheta}\ (\mathrm{T}\vee_2)$$

is more problematic, since standard resolution has no rules that would require unification of two literals in one clause with two literals in another clause. Even propositional intuitionistic logic gives us some surprises. An example is the rule

$$\frac{\Gamma,\ \mathrm{T}\,A,\ \mathrm{F}\,A\supset B}{\Gamma\theta,\ \mathrm{F}\,A\supset B}\ (\mathrm{F}\ \supset_{lr})$$

which again requires to apply a rule based on *two* occurrences of formulas in the same sequent. If we try to simulate this rule by a seemingly appropriate clause

$$\neg N_{\mathrm{T}\,A},\ N_{\mathrm{F}\,A\supset B},$$

we will get an unsound calculus.

Other sequent calculus rules that cannot be captured by standard hyperresolution are the rules with the eigenvariable conditions.

Several "resolution" calculi described in the literature, for example [Mints 1990, Tammet 1996], are in fact inverse calculi in which inference rules are presented as special kind of "input clauses". The inference rules of such calculi are modifications of the standard resolution rule, in which one of the parents is an input clause.

For example, one of the rules in [Mints 1990] is

$$\frac{p, q \Rightarrow r\ \text{(input)}\ ;\quad U \rightarrow p\ ;\quad V \rightarrow q}{U, V \Rightarrow r}$$

which is essentially the $(\mathrm{F}\wedge)$ rule of the inverse calculus.

5.6. Further issues to naming

Names give a very concise representation of subformulas of the goal formula. Since the names are simply atomic formulas, they are easier to handle than arbitrary formulas. There are variations of the naming technique, let us describe two of them here.

5.6.1. No axioms

The rules for forming the set Ax_G corresponding to the axioms of the inverse method calculus can be replaced by less complicated ones, denoted Ax'_G. Instead of building these axioms by unifying every pair of literals, we can replace Ax_G by the set of clauses obtained as follows.

For every free subformula L of G such that L is a literal do the following. Take the name A of L and add the clause $\overline{L}, A$ to Ax'_G.

Consider Example 5.3 on page 225 again. If we use this naming scheme, instead of clauses (1) and (2) of that example we obtain the following three clauses:

$$\begin{array}{ll} (a) & \neg P(z,x), C(z,x). \\ (b) & P(z,fx), E(z,x). \\ (c) & P(x,ffc), F(x). \end{array}$$

There is no more exact correspondence between the hyperresolution derivations from the transformed set of clauses and inverse method derivations. However, it is easy to see that each member of Ax_G can be constructed in one hyperresolution inference between two clauses in Ax'_G resolving two literals of the original signature. For example, clause (1) is obtained by the following hyperresolution inference from the clauses (a) and (b).

$$\frac{\neg P(z,x), C(z,x) \quad P(z,fx), E(z,x)}{C(z,fx),\ E(z,x)}\ (hyper)$$

We can obtain a near exact correspondence between the inverse method derivations and ordered hyperresolution derivations for an ordering in which the atoms of the old signature are greater that the new names.

5.6.2. Formula matching

Suppose that the goal G has two free subformulas F_1 and F_2 such that $F_2 = F_1\sigma$ for some substitution σ. Further, let A_1, A_2 be the names of F_1, F_2 respectively. Then one can replace A_2 by $A_1\sigma$ in all clauses. This optimization can be used for either variant of the naming transformation, using Ax_G or Ax'_G.

Consider again Example 5.3 on page 225. One can see that the free subformula $\neg P(x,ffc)$ is an instance of the free subformula $\neg P(z,fx)$ with the substitution $\{z \mapsto x, x \mapsto fc\}$. Therefore, instead of the name $F(x)$ of $P(x,ffc)$, one can use $E(z,x)\{z \mapsto x, x \mapsto fc\} = E(x,fc)$. For the naming transformation with Ax_G, this would result in the replacement of clauses (2) and (7) by

$$\begin{array}{ll} (2') & C(z,ffc),\ E(z,fc). \\ (7') & \neg E(z,x),\ \neg E(x,fc),\ D(z,x). \end{array}$$

For the naming transformation with Ax'_G, this would result in the replacement of clauses (2) and (c) by $(2')$ and

$$(c')\quad P(x,ffc), E(x,fc).$$

After this the clause (c') becomes an instance of the clause (b) and can be removed.

One can give more complex examples, with non-literal subformulas F_1 and F_2. In principle, for this translation to be valid it is enough to know that F_2 is *equivalent* to $F_1\sigma$, but the problem of the existence of such a substitution σ is undecidable. Moreover, even for a fixed σ, the equivalence of $F_1\sigma$ and F_2 is undecidable. In practice, one can use weaker criteria for equivalence, for example equivalence w.r.t. renaming of bound variables, associativity and commutativity of $\wedge$ and $\vee$.

The naming technique is also important in the development of resolution decision procedures [Fermüller et al. 2001] and in establishing simulation results between various refinements of resolution [Bachmair and Ganzinger 2001] (Chapters 25 and 2 of this Handbook).

5.7. Fast food warning

With your new experience in cooking, you can now try the recipe yourself: take your favorite logic and cook the inverse method calculus for it (or ask your students do so). With most logics you can do it rather quickly and in the most straightforward way. The aim of this section is to warn you: beware of fast food! And you don't go with your students to fast food restaurants, do you? Your meal is not yet delicious: you forgot to season it to perfection. Be patient and wait until you read about seasoning.

6. Season your meal: strategies and redundancies

Recall that all the inverse method calculi designed so far, for example, $\mathbf{K}^G_{inv}$, have the *finite rule property*.

1. Every formula occurring in a derivation of a closed formula G is a subformula of G.
2. There exists a finite number of axioms.
3. For every sequent Γ, there exists only a finite number of ways of applying an inference rule to it (for example, $(\wedge_l)$, $(\wedge_r)$, $(\Diamond)$, $(\Diamond^+)$, and (C) in the case of propositional $\mathbf{K}^G_{inv}$).
4. For every pair of sequents Γ and Δ, there exists only a finite number of ways of applying an inference rule to Γ and Δ (for example, $(\vee)$ in the case of propositional $\mathbf{K}^G_{inv}$).

A typical refutation procedure based on the inverse method uses these properties and works roughly as follows.

1. Let S be a set of sequents, initially $\{p, \neg p \mid p, \neg p$ 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 because the search space (the set of sequents S) grows rapidly, 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 or inference is redundant if it can be omitted without loosing completeness. Very powerful techniques for proving redundancies have been developed in resolution-based calculi for classical logic, see e.g., [Bachmair and Ganzinger 2001, Nieuwenhuis and Rubio 2001] (Chapters 2 and 7 of this Handbook).

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 $\{p, \neg p \mid p, \neg p$ 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 et al. 1996]. This paper is interesting because it discusses many redundancy criteria, but proofs of many statements are missing. In particular, [Mints et al. 1996] use some strategies *deleting* clauses from the search space, though no general technique for proving completeness of calculi with deletion is developed.

To prove completeness of calculi with redundancies in resolution-based theorem proving for *classical* logic, a rather general model-theoretic technique has been developed. Completeness of nearly all known redundancy criteria can be inferred from very general statements about redundant inferences and clauses. However, such a model-theoretic technique was not developed for nonclassical logics. Two different proof-theoretic techniques for nonclassical logics are presented in [Voronkov 1992] and [Voronkov 2001, Voronkov 2000*b*]. In the following sections we will explain the technique of [Voronkov 2001, Voronkov 2000*b*] in detail.

The idea is based on proving properties of refutations in the tableau calculus $\mathbf{K}_{tab}$ and transforming them into redundancy criteria for the inverse method using a suitable Subsequent Lemma. However, the calculus $\mathbf{K}_{tab}$ by itself is not very useful for proving properties of derivations, as many properties are best formulated in terms of occurrences of subformulas in the goal formula. To have a suitable language for speaking about subformulas, we give a reformulation of the sequent calculus as a *path calculus.*

7. Path calculi

In this section we define ***path calculi*** obtained from the inverse method calculi by using *paths* instead of subformulas. Paths encode the position of subformulas in the goal formula. The use of path is similar to the use of the naming technique, but paths (unlike names) contain a lot of information about the occurrence of a subformula in the goal formula. Most importantly, paths encode sequences of inferences needed to obtain the goal formula from subformulas. Moreover, as we show in the next section, the notion of paths allows one to formulate redundancies in sequent calculi in a simple way.

In this section we will introduce path calculi only for propositional **K**, the generalization to other logics with the subformula property is straightforward. The material of this section is based upon [Voronkov 2000*b*, Voronkov 2001]. We will first introduce a tableau-style path calculus for **K**, and then an inverse method calculus. In this section we assume all formulas are in negation normal form.

7.1. The tableau path calculus

We recall the tableau calculus $\mathbf{K}_{tab}$ for propositional **K** presented in Section 4.2.1. It will be helpful to observe the following properties of $\mathbf{K}_{tab}$.

7.1. Lemma. *(i)* $\mathbf{K}_{tab}$ *has the* ***subformula property****: every formula occurring in a derivation of a sequent* Γ *consists of subformulas of formulas of* Γ*. (ii) If a sequent* Γ *has a refutation, then any sequent* Γ, Δ *has a refutation too. (iii) The rules* $(\wedge)$ *and* $(\vee)$ *are invertible.*

In this and the next section G will always denote the ***goal formula***.

7.2. Definition *(path)*. 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.

1. The empty path ϵ is a G-path, and $G|_\epsilon$ is G.
2. Suppose that π is a G-path and $G|_\pi = F$. Then
   (a) 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***.
   (b) 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***.
   (c) 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***.
   (d) 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***.

A path π is 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.

We will use the formula

$$E = \Box P \wedge \Box(\neg P \vee Q) \wedge \Diamond\neg Q \wedge \Diamond Q \tag{7.1}$$

for nearly all examples of this section. We consider conjunction and disjunction right-associative, so, for example, formula (7.1) denotes $\Box P \wedge (\Box(\neg P \vee Q) \wedge (\Diamond\neg Q \wedge \Diamond Q))$. The notion of subformula on a path is illustrated in Figure 23.

Evidently, for each subformula F of G, there is a G-path π such that $G|_\pi = F$. The tableau calculus $\mathbf{K}_{tab}$ used for refuting a formula G can be turned into a calculus where the 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 $\Pi = \pi_1, \ldots, \pi_n$ we denote by $\Pi\Box$ the path sequent $\pi_1\Box, \ldots, \pi_n\Box$. The ***path calculus*** $\mathbb{P}^G_{tab}$ and a refutation in it are given in Figures 24 and 25.

The following *bisimulation lemma* is the key to proving completeness of the path calculus and various refinements thereof. Let us call the ***formula image*** of a path sequent or a derivation in $\mathbb{P}^G_{tab}$ the figure obtained from this path sequent or derivation by replacing every path π by the formula $G|_\pi$.

7.3. LEMMA (Bisimulation Lemma for $\mathbb{P}^G_{tab}$). *(i) Let D be a derivation in $\mathbb{P}^G_{tab}$. Then the formula image of D is a derivation of G in $\mathbf{K}_{tab}$. (ii) Let D' be a derivation of a sequent $A_1, \ldots, A_n$ in $\mathbf{K}_{tab}$, and $\pi_1, \ldots, \pi_n$ be positions such that $G|_{\pi_i} = A_i$, for all $i = 1, \ldots, n$. Then there exists a derivation D of $\pi_1, \ldots, \pi_n$ in $\mathbb{P}^G_{tab}$ such that D' is the formula image of D. (iii) Both (i) and (ii) also hold if "derivation" is replaced by "refutation" everywhere.*

PROOF. (i) is proved by the straightforward analysis of inferences in $\mathbb{P}^G_{tab}$. (iii) easily follows from (i) and (ii). We will only show how to prove (ii). The proof is by induction on the number of inferences in D'. We consider only one case, when the last inference is $(\vee)$, other cases are similar. In this case without loss of generality we assume that $A_n = B \vee C$ and that A_n is the principal formula of the last inference. Then the derivation D' has the form

$$\frac{\begin{array}{c}\vdots\, D'_1 \\ A_1, \ldots, A_{n-1}, B\end{array} \quad \begin{array}{c}\vdots\, D'_2 \\ A_1, \ldots, A_{n-1}, C\end{array}}{A_1, \ldots, A_{n-1}, B \vee C}\ (\vee).$$

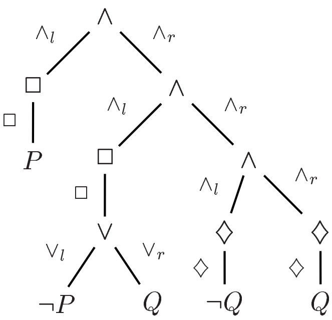
Evidently, $\pi_n\vee_l$ and $\pi_n\vee_r$ are paths in G. Applying the induction hypothesis to the derivations D'_1 and D'_2, we obtain derivations D_1 and D_2 such that D'_1 and D'_2 are their formula images. Define D as the derivation

$$\frac{\begin{array}{c}\vdots\, D_1 \\ \pi_1, \ldots, \pi_{n-1}, \pi_n\vee_l\end{array} \quad \begin{array}{c}\vdots\, D_2 \\ \pi_1, \ldots, \pi_{n-1}, \pi_n\vee_r\end{array}}{\pi_1, \ldots, \pi_{n-1}, \pi_n}\ (\vee).$$

It is not hard to argue that D' is the formula image of D. $\Box$

As a corollary of the Bisimulation Lemma and completeness of $\mathbf{K}_{tab}$, we obtain

7.4. THEOREM (Completeness of $\mathbb{P}^G_{tab}$). *A formula G is unsatisfiable if and only if ϵ has a refutation in $\mathbb{P}^G_{tab}$.*



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

Figure 23: Paths and subformulas in the formula E

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

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

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 24: Path sequent calculus $\mathbb{P}^G_{tab}$

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

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

Figure 25: Refutation in $\mathbb{P}^E_{tab}$

PROOF. By the completeness of $\mathbf{K}_{tab}$, unsatisfiability of G is equivalent to the existence of a refutation of G in $\mathbf{K}_{tab}$.

$\Rightarrow$ Let ϵ have a refutation $\mathbb{P}^G_{tab}$. By the Bisimulation Lemma there exists a refutation in $\mathbf{K}_{tab}$ of the formula $G|_\epsilon$, i.e., of G.

$\Leftarrow$ Let D' be a refutation of G in $\mathbf{K}_{tab}$. By the Bisimulation Lemma there exists a refutation D in $\mathbb{P}^G_{tab}$ whose formula image is D'. Evidently, D is a refutation of ϵ.

□

7.2. Proof-theoretic properties of the tableau path calculus

We will now prove several statements about properties of derivations in the tableau path calculus $\mathbb{P}^G_{tab}$, and then use them to find redundancies in the inverse path calculus defined below in Section 7.3.

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.

7.5. LEMMA. *Let Π be a path sequent occurring in a derivation of ϵ in $\mathbb{P}^G_{tab}$. Then all paths in Π have the same modal length.*

PROOF. First, observe the following fact. Let an inference in $\mathbb{P}^G_{tab}$ 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. □

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.

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

PROOF. As in the proof of Lemma 7.5, observe that this property holds for all premises of an inference, whenever it holds for its conclusion. □

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 $\pi'_1, \pi'_2, \pi''_1, \pi''_2$ such that the paths π'_1 and π'_2 have the same modal length and such that $\pi'_1 \neq \pi'_2$.

For example, consider the paths of Figure 23. The paths $\wedge_r \wedge_l \Box \vee_l$ 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.

7.7. LEMMA. *Let* Π *be a path sequent occurring in a derivation of* ϵ *in* $\mathbb{P}^G_{tab}$. *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

$$\begin{array}{c} \dfrac{\Gamma_1, \pi\vee_l \quad \Gamma_1, \pi\vee_r}{\Gamma_1, \pi}\ (\vee) \\ \vdots \\ \dfrac{\Gamma_2, \pi\vee_l \qquad\qquad \Gamma_2, \pi\vee_r}{\Gamma_2, \pi}\ (\vee). \end{array}$$

Since Γ_1, π is above $\Gamma_2, \pi\vee_r$ in the derivation, Γ_2 must contain a prefix of π. This contradicts Lemma 7.6, since the path sequent $\Gamma_2, \pi\vee_r$ should be prefix-free. $\Box$

7.8. LEMMA. *Let* Π *be a path sequent occurring in a derivation of* ϵ *in* $\mathbb{P}^G_{tab}$. *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

$$\begin{array}{c} \dfrac{\Gamma_1\Box, \pi_1'\Diamond}{\Gamma_1, \pi_1', \Delta_1}\ (\Diamond) \\ \vdots \\ \dfrac{\Gamma_2\Box, \pi_2'\Diamond}{\Gamma_2, \pi_2', \Delta_2}\ (\Diamond). \end{array}$$

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 Lemma 7.5, since all paths in a path sequent must have the same modal length. $\Box$

In view of Lemmas 7.7 and 7.8 we say that a pair of paths π_1, π_2 are ***in conflict*** if they either form a $\vee$-fork or are diamond-separated.

The lemmas above assert some properties of all possible derivations in the path calculus. Now we want to restrict search for a 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 equivalent, called a *G-ordering*. 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 25:

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

It is essential in this 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 ***brothers*** if one of them has the form $\pi\wedge_l$ and the other $\pi\wedge_r$, or likewise, if 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 brothers: $\wedge_r\wedge_l\Box\vee_l$ 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 brothers.

We denote by the generic symbol $⩙$ any of the symbols $\wedge$, $\vee$. Likewise, we denote by $*$ any of the symbols l, r and use notation like $\wedge_*$ or $\vee_*$. Thus $⩙_*$ denotes any of the symbols $\wedge_l$, $\wedge_r$, $\vee_l$, $\vee_r$. We also denote by the generic symbol $\Box\!\!\!\!\Diamond$ 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 length of π_1 is strictly greater than the modal length of π_2; or
   (b) π_1, π_2 have the same modal length, the last symbol of π_1 is $⩙_*$, and the last symbol of π_2 is $\Box\!\!\!\!\Diamond$; or
   (c) π_1, π_2 have the same modal length and $\pi_2 \sqsubset \pi_1$. (Note that this case and the previous one are not exclusive, but they do not contradict each other.)
2. There is no path between two brothers, i.e., there exists no G-paths π_1, π_2, π_3 such that $\pi_1 \succ \pi_2 \succ \pi_3$ and π_1, π_3 are brothers.

The intuitive meaning of $\succ$ is that we should first apply rules to the formulas greater with respect to $\succ$. The relation $\succ$ is also defined in such a way that the

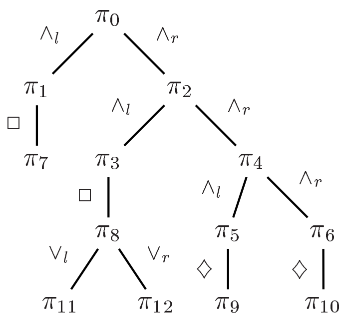


Figure 26: A E-ordering

conclusion of any rule is less than both its premises in the multiset ordering on sequents induced by $\succ$. Condition 1a ensures that the conclusion is less than the premise when the rules $(\Diamond)$ or $(\Diamond^+)$ are applied. Condition 1b is due to the fact that an application of $(\Diamond)$ or $(\Diamond^+)$ to a sequent containing a path $\pi \Box\!\!\!\!\Diamond_*$ may give an incomplete calculus. Conditions 1c and 2 are more technical and used in some proofs of this section.

It will be clear why these conditions are required of G-orderings when we prove Lemma 7.12 below.

To illustrate G-orderings, we will show one possible E-ordering (recall that E is our example goal formula). The paths in E were illustrated in Figure 23. In Figure 26, we show these paths again; this time each path is labeled as π_i, where i is a natural number. We claim that the ordering given by $\pi_i \succ \pi_j$ if $i > j$ is an E-ordering. Indeed, the conditions (1a)–(1c) of G-orderings are easy to check. We leave it to the reader to verify that condition (2) holds, too.

We write $\pi \succeq \pi'$ to express that $\pi \succ \pi'$ or $\pi = \pi'$.

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. In the modal case, we want also to 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.* This obstacle leads to a slightly complex definition of the ordering on paths.

We will now prove that G-orderings exist for every G. To this end we will describe an ***ordering procedure*** that orders G-paths and note that the resulting ordering

Initially, define S_i to be the set of paths in G of modal length i; /* 1 */
<u>for all</u> S_i except the last one **<u>do</u>** /* 2 */
  <u>let</u> $\pi_1, \ldots, \pi_n$ be all paths in S_i ending at $\Box\!\!\!\Diamond$; /* 3 */
  split S_i into $S_i - \{\pi_1, \ldots, \pi_n\} \succ \{\pi_1\} \succ \ldots \succ \{\pi_n\}$; /* 4 */
split S_0 into $S_0 - \{\epsilon\} \succ \{\epsilon\}$; /* 5 */
<u>while</u> there exists S_i with more than one member **<u>do</u>** /* 6 */
  <u>let</u> $\pi\bowtie_l, \pi\bowtie_r$ be two brothers in S_i such that $\pi \notin S_i$; /* 7 */
  <u>let</u> $L = \{\pi' \mid \pi' \in S_i$ and $\pi\bowtie_l \sqsubset \pi'\}$;
  <u>let</u> $R = \{\pi' \mid \pi' \in S_i$ and $\pi\bowtie_r \sqsubset \pi'\}$; /* 8 */
  split S_i into $S_i - (R \cup L \cup \{\pi\bowtie_r, \pi\bowtie_l\}) \succ R \succ L \succ \{\pi\bowtie_r\} \succ \{\pi\bowtie_l\}$; /* 9 */

Figure 27: An ordering procedure

is a G-ordering.

The procedure works with a sequence of sets of paths; this sequence is denoted $S_n \succ S_{n-1} \succ \ldots \succ S_0$. The intuitive meaning of the sequence is that for all $\pi \in S_i$ and $\pi' \in S_{i-1}$ we have $\pi \succ \pi'$. The paths belonging to the same S_i are yet unordered (but will be ordered later). At each step we will pick up one set S_i in the sequence containing one or more members and replace S_i by two or more sets $S_{i1} \succ \ldots \succ S_{ik}$ such that $S_{i1} \cup \ldots \cup S_{ik} = S_i$. The procedure terminates when each set contains exactly one member. An ordering procedure is shown in Figure 27. Note that some sets obtained by the procedure (for example L or R) may be empty; in this case we do not include them in the sequence. When the procedure terminates, the sequence consists of singleton sets. We then let $\pi \succ \pi'$ if the sequence has the form $\ldots \{\pi\} \succ \ldots \succ \{\pi'\} \ldots$.

We show how the E-ordering of Figure 26 can be obtained by this procedure. The splitting steps will be denoted by $\rightarrow$ superscribed by the line number in the algorithm. Instead of singleton sets $\{\pi_i\}$ we write π_i.

$$
\begin{array}{l}
\{\pi_7, \pi_8, \pi_9, \pi_{10}, \pi_{11}, \pi_{12}\} \succ \{\pi_0, \pi_1, \pi_2, \pi_3, \pi_4, \pi_5, \pi_6\} \rightarrow^4 \\
\{\pi_{11}, \pi_{12}\} \succ \pi_{10} \succ \pi_9 \succ \pi_8 \succ \pi_7 \succ \{\pi_0, \pi_1, \pi_2, \pi_3, \pi_4, \pi_5, \pi_6\} \rightarrow^5 \\
\{\pi_{11}, \pi_{12}\} \succ \pi_{10} \succ \pi_9 \succ \pi_8 \succ \pi_7 \succ \{\pi_1, \pi_2, \pi_3, \pi_4, \pi_5, \pi_6\} \succ \pi_0 \rightarrow^9 \\
\pi_{12} \succ \pi_{11} \succ \pi_{10} \succ \pi_9 \succ \pi_8 \succ \pi_7 \succ \{\pi_1, \pi_2, \pi_3, \pi_4, \pi_5, \pi_6\} \succ \pi_0 \rightarrow^9 \\
\pi_{12} \succ \pi_{11} \succ \pi_{10} \succ \pi_9 \succ \pi_8 \succ \pi_7 \succ \{\pi_3, \pi_4, \pi_5, \pi_6\} \succ \pi_2 \succ \pi_1 \succ \pi_0 \rightarrow^9 \\
\pi_{12} \succ \pi_{11} \succ \pi_{10} \succ \pi_9 \succ \pi_8 \succ \pi_7 \succ \{\pi_5, \pi_6\} \succ \pi_4 \succ \pi_3 \succ \pi_2 \succ \pi_1 \succ \pi_0 \rightarrow^9 \\
\pi_{12} \succ \pi_{11} \succ \pi_{10} \succ \pi_9 \succ \pi_8 \succ \pi_7 \succ \pi_6 \succ \pi_5 \succ \pi_4 \succ \pi_3 \succ \pi_2 \succ \pi_1 \succ \pi_0.
\end{array}
$$

7.9. LEMMA. *Every ordering produced by an ordering procedure on G is a G-ordering.*

The proof can be found in [Voronkov 2001].

Since any run of the ordering procedure results in a G-ordering, we obtain the following corollary.

7.10. COROLLARY. *G-orderings exist.* □

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 $(\vee)$-inference in $\mathbb{P}^G_{tab}$

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

is said to ***respect a G-ordering*** $\succ$ if $\pi\vee_l \succ \Gamma$ and $\pi\vee_r \succ \Gamma$. Likewise, a $(\wedge)$-inference in $\mathbb{P}^G_{tab}$

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

is said to respect $\succ$ if $\pi\wedge_l \succ \Gamma$ and $\pi\wedge_r \succ \Gamma$. A derivation in $\mathbb{P}^G_{tab}$ is said to respect $\succ$ if every $(\vee)$ and every $(\wedge)$-inference of this derivation respect $\succ$.

7.11. LEMMA. *Let G be a unsatisfiable formula and $\succ$ be a G-ordering. Then there exists a refutation of ϵ in $\mathbb{P}^G_{tab}$ that respects $\succ$.*

It is not immediately obvious how to prove this lemma. If we try to prove this lemma by induction, starting with the empty 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 $\succ$. Let us illustrate such a deadlock situation. Suppose that $\pi_1\Box$ and $\pi_2\vee_l$ are G-paths such that π_1, π_2 have the same modal length and $\pi_1 \succ \pi_2\vee_l$. Consider the sequent π_1, π_2. We can apply a $(\vee)$-inference in the tableau path calculus to obtain this sequent:

$$\frac{\pi_1, \pi_2\vee_l \quad \pi_1, \pi_2\vee_r}{\pi_1, \pi_2}\ (\vee).$$

In fact, this inference is *the only* way to derive π_1, π_2. However, this inference 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.

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 $\bowtie$-path π_i in Γ we have $\pi_i\bowtie_* \succ \Gamma$." Note that $\pi_i\bowtie_* \succ \pi_i$ is guaranteed by the conditions on G-orderings, therefore the requirement $\pi_i\bowtie_* \succ \Gamma$ can be replaced by $\pi_i\bowtie_* \succ \Gamma \setminus \{\pi_i\}$.

The next lemma asserts that a compact sequent Γ cannot lead to a deadlock.

7.12. LEMMA. *Let $\succ$ be a G-ordering. Then*

1. *the premise of every $(\Diamond)$-inference is $\succ$-compact.*
   Furthermore, let Γ be a $\succ$-compact sequent occurring in a derivation of ϵ. Then
2. *every $(\bowtie)$-inference having Γ as the conclusion, respects $\succ$;*
3. *if Γ contains at least one $\wedge$-path or $\vee$-path, then there exists a $(\bowtie)$-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 $\pi'\bowtie_*$ is a G-path, then $\pi' \succ \Pi\Box, \pi\Diamond - \{\pi'\}$. Let us prove $\pi\Diamond\bowtie_* \succ \Pi\Box$, the case when $\pi' \in \Pi\Box$ is similar. By Lemma 7.5, all paths in $\Pi\Box, \pi\Diamond$ are of the same modal length, so $\pi\Diamond\bowtie_*$ is of the same modal length as any path in $\Pi\Box$. By condition 1b of the definition of G-ordering we have $\pi\Diamond\bowtie_* \succ \Pi\Box$.
2. We prove: *every $(\bowtie)$-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 $\bowtie$-path and prove: *there exists a $(\bowtie)$-inference all whose premises are $\succ$-compact.* Consider the set of paths

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

   By the assumption, this set is nonempty. Since $\succ$ is a linear ordering, this set has the least element with respect to $\succ$. 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) $\pi \wedge_* \wedge\!\!\!\vee_* \succ \Pi$, whenever $\pi \wedge_* \wedge\!\!\!\vee_*$ is a G-path; (3b) for every path $\pi' \in \Pi$ we have $\pi' \wedge\!\!\!\vee_* \succ \Pi, \pi\wedge_*$, whenever $\pi' \wedge\!\!\!\vee_*$ is a G-path; and (3c) $\pi \wedge_l \wedge\!\!\!\vee_* \succ \pi\wedge_r$ and $\pi \wedge_l \wedge\!\!\!\vee_* \succ \pi\wedge_r$. Let us prove this.

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

(b) Since Π, π is compact, $\pi' \wedge\!\!\!\vee_* \succ \Pi$, so it remains to prove $\pi' \wedge\!\!\!\vee_* \succ \pi\wedge_*$, i.e. $\pi' \wedge\!\!\!\vee_* \succ \pi\wedge_l$ and $\pi' \wedge\!\!\!\vee_* \succ \pi\wedge_r$. Note that $\pi' \wedge\!\!\!\vee_* \in S$ and that $\pi\wedge_l$ is the least element of S, therefore $\pi' \wedge\!\!\!\vee_* \succeq \pi\wedge_l$. By Prefix-Free Lemma 7.6, $\pi' \wedge\!\!\!\vee_*$ and $\pi\wedge_l$ cannot coincide, thus $\pi' \wedge\!\!\!\vee_* \succ \pi\wedge_l$. Again by this lemma, $\pi' \wedge\!\!\!\vee_*$ cannot be a prefix of $\pi\wedge_l$. In addition, we have $\pi' \wedge\!\!\!\vee_* \succ \pi\wedge_l$, and $\pi\wedge_l$ is a brother to $\pi\wedge_r$, then by condition 2 of the definition of G-ordering, $\pi' \wedge\!\!\!\vee_* \succ \pi\wedge_r$.

(c) We will only prove $\pi \wedge_l \wedge\!\!\!\vee_* \succ \pi\wedge_r$, the second statement is symmetric. Suppose, by contradiction, $\pi\wedge_r \succ \pi \wedge_l \wedge\!\!\!\vee_*$. Since $\succ$ respects the prefix relation, we also have $\pi\wedge_r \succ \pi \wedge_l \wedge\!\!\!\vee_* \succ \pi\wedge_l$. This contradicts condition 2 of the definition of G-orderings, since the path $\pi \wedge_l \wedge\!\!\!\vee_*$ is between two brothers.

□

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 $\mathbf{K}_{tab}$ is greater than its conclusion.

Now we can prove Lemma 7.11: if G is unsatisfiable, then there exists a refutation of ϵ in $\mathbb{P}^G_{tab}$ that respects $\succ$.

PROOF (*of Lemma 7.11*). 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 7.12 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 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 7.12 there exists an ($\wedge\!\!\!\vee$)-inference having Π as the conclusion and $\succ$-compact premises. Since ($\wedge\!\!\!\vee$)-rules are invertible (Lemma 7.1), 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 $\mathbf{K}_{tab}$ with the conclusion Γ and unsatisfiable premise Γ'. It is not hard to argue that there exists an ($\Diamond$)-inference in $\mathbb{P}^G_{tab}$ with a

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

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

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

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

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 28: Inverse path calculus $\mathbb{P}^G_{inv}$

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

Some sequences of inferences are shown as one inference.

Figure 29: Derivation in $\mathbb{P}^G_{inv}$

premise Π' such that Γ' is the formula image of Π'. By Lemma 7.12, Π' is $\succ$-compact.

The termination of this construction follows from the following observation: every infinite construction would yield an infinite derivation in $\mathbf{K}_{tab}$ which does not exist. $\Box$

7.3. Inverse path calculus

In this section we define the inverse version of the path calculus, obtained by turning $\mathbf{K}^G_{inv}$ into a path calculus.

The ***inverse path calculus*** $\mathbb{P}^G_{inv}$ and a refutation in it are given in Figures 28 and 29.

The same arguments as in Bisimulation Lemma 7.3 prove the following lemma.

7.13. LEMMA (Bisimulation Lemma for $\mathbb{P}^G_{inv}$). *(i) Let D be a derivation in $\mathbb{P}^G_{inv}$. Then the formula image of D is a derivation of G in $\mathbf{K}^G_{inv}$. (ii) Let D' be a derivation* *of a sequent $A_1, \ldots, A_n$ in $\mathbf{K}^G_{inv}$, and $\pi_1, \ldots, \pi_n$ be paths such that $G|_{\pi_i} = A_i$, for all $i = 1, \ldots, n$. Then there exists a derivation D of $\pi_1, \ldots, \pi_n$ in $\mathbb{P}^G_{inv}$ such that D' is the formula image of D. (iii) Both (i) and (ii) also hold if "derivation" is replaced by "refutation" everywhere.* □

The calculus $\mathbb{P}^G_{inv}$ has different rules for handling $\wedge$, so we change the definition of derivations respecting a G-ordering in the following way.

A derivation in $\mathbb{P}^G_{inv}$ is said to ***respect a G-ordering*** $\succ$ if

1. for every ($\vee$) inference of this derivation

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

   we have $\pi\vee_l \succ \Gamma$ and $\pi\vee_r \succ \Delta$;
2. likewise, for every ($\wedge_l$) inference (respectively ($\wedge_r$) inference) of this derivation

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

   we have $\pi\wedge_l \succ \Gamma$ (respectively $\pi\wedge_r \succ \Gamma$).

The next theorem is the main theorem of this section. 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 $\mathbb{P}^{G,\succ}_{inv}$ the calculus obtained from $\mathbb{P}^G_{inv}$ 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 $\succ$.

7.14. THEOREM (Completeness of $\mathbb{P}^{G,\succ}_{inv}$). *A formula G is unsatisfiable if and only if ϵ has a refutation in the calculus $\mathbb{P}^{G,\succ}_{inv}$.*

The key lemma in the proof of Theorem 7.14 is as follows.

7.15. LEMMA (Subsequent Lemma for $\mathbb{P}^{G,\succ}_{inv}$). *Suppose that D is a refutation of ϵ in $\mathbb{P}^G_{tab}$ that respects $\succ$ and I is an inference in D of the form*

$$\frac{\Gamma_1 \quad \cdots \quad \Gamma_n}{\Gamma},$$

and path sequents $\Delta_1, \ldots, \Delta_n$ are such that each Δ_i is a subsequent of Γ_i. Then there exists a derivation D' of Δ in $\mathbb{P}^{G,\succ}_{inv}$ from $\Delta_1, \ldots, \Delta_n$ such that Δ is a subsequent of Γ.

PROOF. We will give a proof by case analysis on the inference I, after having noted one general property of $\Delta, \Delta_1, \ldots, \Delta_n$.

By Lemmas 7.7 and 7.8 each path sequent among $\Gamma_1, \ldots, \Gamma_n$ contains no conflicting pair of paths. Since $\Delta_1, \ldots, \Delta_n$ are submultisets of $\Gamma_1, \ldots, \Gamma_n$, respectively, we obtain that none of $\Delta_1, \ldots, \Delta_n$ contains a conflicting pair of paths. Likewise, if we find a derivation of Δ from $\Delta_1, \ldots, \Delta_n$, since Δ is a submultiset of Γ, we obtain that Δ contains no conflicting pair of paths. In a similar way we can check that it contains no path sequent with a pair of paths of different modal length. Therefore, it is enough to check that the intermediate sequents of D' do not violate the condition, which will be obvious in all cases.

Therefore, we will only show how to find the derivation D' and prove that it respects $\succ$. Consider all possible cases of inferences in $\mathbb{P}^G_{tab}$.

1. The inference I is an axiom Γ, π_1, π_2 such that $G|_{\pi_1} = \neg G|_{\pi_2}$. Then π_1, π_2 is a subsequent of Γ, π_1, π_2, and is an axiom of $\mathbb{P}^{G,\succ}_{inv}$. Therefore, we can let D' be the derivation consisting of one axiom π_1, π_2. Evidently, it respects $\succ$.
2. The inference I has the form

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

Note that $\pi\wedge_l, \pi\wedge_r \succ \Pi$, because D respects $\succ$. We assume that some sequent Δ_1 is a subsequent of $\Pi, \pi\wedge_l, \pi\wedge_r$ and find a derivation in $\mathbb{P}^G_{inv}$ of a subsequent of Π, π from Δ_1.

Consider the following possible cases:

(a) *Case: Δ_1 contains neither $\pi\wedge_l$, nor $\pi\wedge_r$.* In this case Δ_1 is a subsequent of Π. Then Δ_1 is also a subsequent of Π, π, and the empty derivation of Δ_1 from Δ_1 proves the case.

(b) *Case: Δ_1 contains $\pi\wedge_l$, but not $\pi\wedge_r$.* In this case $\Delta_1 = \Delta', \pi\wedge_l$ for some subsequent Δ' of Π. The following derivation proves the case:

$$\frac{\Delta', \pi\wedge_l}{\Delta', \pi}\ (\wedge_l).$$

Since $\pi\wedge_l, \pi\wedge_r \succ \Pi$ and Δ' is a submultiset of Π, we have $\pi\wedge_l \succ \Delta'$. The ordering condition can be checked in the similar way for all other cases, we do not consider it any more.

(c) *Case: Δ_1 contains $\pi\wedge_r$, but not $\pi\wedge_l$.* This case is symmetric to the previous one.

(d) *Case: Δ_1 contains both $\pi\wedge_l$ and $\pi\wedge_r$.* In this case $\Delta_1 = \Delta', \pi\wedge_l, \pi\wedge_r$ for some submultiset Δ' of Π. Without loss of generality assume $\pi\wedge_l \succ \pi\wedge_r$ (and hence $\pi\wedge_l \succ \Delta', \pi\wedge_r$). The following derivation proves the case:

$$\cfrac{\cfrac{\cfrac{\Delta', \pi\wedge_l, \pi\wedge_r}{\Delta', \pi, \pi\wedge_r}\ (\wedge_l)}{\Delta', \pi, \pi}\ (\wedge_r)}{\Delta', \pi}\ (C).$$

Other cases are considered in [Voronkov 2001].

□

By induction on the length of derivations one can easily generalize Lemma 7.15 from inferences to arbitrary derivations:

7.16. LEMMA. *Let D be a refutation of ϵ in $\mathbb{P}^G_{tab}$ that respects $\succ$, D'' be a subderivation in D of a sequent Γ from sequents $\Gamma_1, \ldots, \Gamma_n$, and $\Delta_1, \ldots, \Delta_n$ be subsequents of $\Gamma_1, \ldots, \Gamma_n$, respectively. Then there exists a derivation D' of Δ in $\mathbb{P}^{G,\succ}_{inv}$ from $\Delta_1, \ldots, \Delta_n$ such that Δ is a subsequent of Γ.* □

Now we can prove Theorem 7.14.

PROOF (*of Theorem 7.14*).

(⇒) Suppose that a formula G is unsatisfiable. By Completeness Theorem 7.4 for $\mathbb{P}^G_{tab}$, the sequent ϵ has a refutation in $\mathbb{P}^G_{tab}$. By Lemma 7.16, there exists a subsequent of ϵ that has a refutation in $\mathbb{P}^{G,\succ}_{inv}$. Evidently, the empty sequent has no refutation in $\mathbb{P}^G_{inv}$; therefore ϵ has a refutation in $\mathbb{P}^{G,\succ}_{inv}$.

(⇐) Suppose that ϵ has a refutation in $\mathbb{P}^{G,\succ}_{inv}$. By Bisimulation Lemma 7.13 for $\mathbb{P}^G_{inv}$, the sequent G has a refutation in $\mathbf{K}^G_{inv}$. It is not hard to argue that $\mathbf{K}^G_{inv}$ is sound, i.e., every sequent that has a refutation is unsatisfiable. Therefore, G is unsatisfiable.

□

Let $\succ$ be a G-ordering. Consider the standard ***multiset extension*** of $\succ$, denoted $\succ^{mul}$, defined as follows: $\Pi \succ^{mul} \Pi'$ if $\Pi \neq \Pi'$ and for every $\pi' \mathrel{\dot\in} \Pi' \mathbin{\dot-} \Pi$ there exists a $\pi \mathrel{\dot\in} \Pi \mathbin{\dot-} \Pi'$ such that $\pi \succ \pi'$. It is well-known that $\succ^{mul}$ is a well-founded order. Note that all inference rules of $\mathbb{P}^{G,\succ}_{inv}$ have the following property: the conclusion of any rule is strictly less that every premise of this rule with respect to $\succ^{mul}$. For ($\between$) rules, this follows from the fact that the conclusion is obtained from the multiset union of the premises by replacing the greatest path by one or more smaller paths. For the contraction rule this is obvious. For ($\Diamond$) and ($\Diamond^+$)-rules, this follows from the first condition of G-orderings. This implies that any derivation process from a finite set of path sequents in $\mathbb{P}^{G,\succ}_{inv}$ will terminate (so we do not even need a subsumption rule that is often essential for termination of bottom-up derivation processes).

7.4. 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.

In order to formalize subsumption as a redundancy criterion, we need a new notion of derivation, since subsumed clauses are not redundant per se, but only redundant with respect to other clauses in the search space. So 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: (i) inference rules are reformulated as behaving on sets of clauses, as in [Bachmair and Ganzinger 2001] (Chapter 2 of this Handbook); (ii) a very general property of derivations on such sets is proved to guarantee completeness in the style of Voronkov [1992]. 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 chapter 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

$$\frac{\Pi_1 \cdots \Pi_n}{\Pi} \tag{7.2}$$

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

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

The intuitive meaning of $\{\mathcal{I}\}$ is the following: sets of objects formalize search space; the inference rule above means: if $\Pi_1, \dots, \Pi_n$ are already proved (belong to the search space), we can add Π to the search space, provided that (7.2) is an inference of $\mathcal{I}$. Inferences in $\{\mathcal{I}\}$ will be denoted using the $\triangleright$ sign:

$$S \cup \{\Pi_1, \dots, \Pi_n\} \triangleright S \cup \{\Pi_1, \dots, \Pi_n, \Pi\}.$$

Sometimes we will superscribe $\triangleright$ with the name of the inference rule.

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

$$S_0 \triangleright S_1 \triangleright S_2 \triangleright \dots\ ,$$

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:

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

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 $S_0 \triangleright S_1 \triangleright \dots$ in $\{\mathcal{I}\}$. The set $\bigcup_i \bigcap_{j \geq i} S_j$ is called the ***limit*** of this derivation. It is easy to see that the limit consists of the sequents that remain in the derivation forever after some step. The derivation is called ***fair*** if, for every inference of $\mathcal{I}$,

$$\frac{\Pi_1 \cdots \Pi_n}{\Pi} ,$$

if $\Pi_1, \dots, \Pi_n$ belong to the limit of the derivation, then an inference of $\{\mathcal{I}\}$

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

has been applied in the derivation. This means that for some i and S, the set S_i has the form $S \cup \{\Pi_1, \dots, \Pi_n\}$, and S_{i+1} has the form $S \cup \{\Pi_1, \dots, \Pi_n, \Pi\}$.

We will now prove a statement about $\{\mathbb{P}_{inv}^{G,\succ}\}$, i.e., the set inference system obtained from $\mathbb{P}_{inv}^{G,\succ}$. It will be clear from the proof that the statement is very general and applicable to any inference system without the deletion rule.

7.17. THEOREM (Completeness of $\{\mathbb{P}_{inv}^{G,\succ}\}$). *Let G be a formula and $\succ$ be a G-ordering. (i) If some derivation in $\{\mathbb{P}_{inv}^{G,\succ}\}$ succeeds, then G is unsatisfiable. (ii) If G is unsatisfiable, then every fair derivation in $\{\mathbb{P}_{inv}^{G,\succ}\}$ succeeds.*

PROOF. (i) Let a derivation $S_0 \triangleright \dots \triangleright S_n$ succeed. It is easy to see that every $\Pi \in \bigcup_i S_i$ has a refutation in $\mathbb{P}_{inv}^{G,\succ}$. Therefore ϵ has a refutation, and by completeness of $\mathbb{P}_{inv}^{G,\succ}$ (Theorem 7.14), G is unsatisfiable.

(ii) Let $\mathcal{S} = S_0 \triangleright S_1 \triangleright \dots$ be a fair derivation. We prove that every path sequent Π that has a refutation in $\mathbb{P}_{inv}^{G,\succ}$ occurs in some S_i. The proof is by induction on the length of the refutation of Π in $\mathbb{P}_{inv}^{G,\succ}$. If the refutation is of length 0, i.e., Π is an axiom, then Π is already a member of S_0. For refutations of longer lengths, consider the last inference of this refutation:

$$\frac{\Pi_1 \quad \dots \quad \Pi_n}{\Pi} .$$

By the induction hypothesis, every Π_i is a member of some S_{k_i}. Since $\{\mathbb{P}_{inv}^{G,\succ}\}$ does not delete any path sequents, all Π_i belong to the limit of $\mathcal{S}$. Then by fairness of $\mathcal{S}$, Π is a member of some S_m, too. $\square$

Now we are ready to prove that $\{\mathbb{P}^{G,\succ}_{inv}\}$ 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 $\{\mathbb{P}^{G,\succ}_{inv}\}$ is complete with this stronger notion.

We say that a path sequent Π ***prefix-subsumes*** a path sequent Π' if for every $\pi \mathrel{\dot\in} \Pi \mathrel{\dot-} \Pi'$ there exists a $\pi' \mathrel{\dot\in} \Pi' \mathrel{\dot-} \Pi$ such that $\pi \sqsubset \pi'$. It is easy to see that prefix-subsumption is the standard multiset extension of the ordering $\sqsubset$, see [Nieuwenhuis and Rubio 2001, page 383] (Chapter 7 of this Handbook) for precise definitions.

The intuition behind the prefix-subsumption is the following. Suppose that a path $\pi_1\pi_2$ occurs in a path sequent Π, occurring in a derivation of ϵ, and π_2 contains no modal operators. If we consider the branch of this derivation between Π and ϵ, there will be a rule that adds π_1 on the branch. Now, if we replace $\pi_1\pi_2$ in Π by π_1, obtaining a path sequent Π', one can see that the derivation can be made into a (maybe shorter) derivation or ϵ from Π'. Therefore, replacement of a path by its subpath of the same modal length preserves derivability of ϵ, and sometimes make derivations shorter.

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

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

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

Note that the notion of fair derivation introduced above only required inferences of the original calculus (*cumulative* inferences) be applied at some step. Now we have new kinds of steps that *delete* sequents from the search space. We will use the old notion of fairness for $\{\mathbb{P}^{G,\succ}_{inv}\}$ with prefix-subsumption, i.e. we do *not* require that prefix-subsumption inferences applicable to sequents in the limit be applied at some step. Thus, any fair derivation in $\{\mathbb{P}^{G,\succ}_{inv}\}$ without prefix-subsumption is also a fair derivation in $\{\mathbb{P}^{G,\succ}_{inv}\}$ with prefix-subsumption.

Let us note some obvious properties of prefix-subsumption:

7.18. Lemma.

1. *Note that* Π *subsumes* Π' *if and only if* $\Pi \mathrel{\dot-} \Pi'$ *is empty. Therefore, if* Π *subsumes* Π'*, then* Π *also prefix-subsumes* Π'*.*
2. *If* Π_1 *prefix-subsumes* Π_2 *and* Π_2 *prefix-subsumes* Π_1*, then* $\Pi_1 = \Pi_2$.[7].
3. *If* Π *prefix-subsumes* Π' *and* Π' *prefix-subsumes* Π''*, then* Π *prefix-subsumes* Π''*.*
4. *If* $\pi_1 \sqsubset \pi'$*, ..., * $\pi_n \sqsubset \pi'$*, and* Π *prefix-subsumes* Π'*, then* $\Pi, \pi_1, \dots, \pi_n$ *prefix-subsumes* Π', π'*.*
5. *If* $\pi \sqsubseteq \pi'$ *and* Π *prefix-subsumes* Π'*, then* Π, π *prefix-subsumes* Π', π'*.*

[7]Using sets instead of multisets in the formulation of the inverse method creates a subtle problem with prefix-subsumption. The path sequents $\pi\pi', \pi$ and $\pi\pi'$ would prefix-subsume each other if we used sets instead of multisets (because $\pi\pi'$ prefix-subsumes $\pi\pi', \pi$, which, in turn, prefix-subsumes $\pi\pi', \pi\pi'$).

6. *There is no infinite sequence* $\Pi_0, \Pi_1, \ldots$ *such that* Π_{i+1} *prefix-subsumes* Π_i *and* $\Pi_{i+1} \neq \Pi_i$.

Many of these properties immediately follow from the standard facts about the multiset extensions of well-founded orders, see [Nieuwenhuis and Rubio 2001, page 383] (Chapter 7 of this Handbook); for example the prefix-subsumption relation is a well-founded order because so is $\sqsubset$.

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

7.19. THEOREM (Completeness of $\{\mathbb{P}^{G,\succ}_{inv}\}$ With Prefix-Subsumption). *(i) If any derivation in the system* $\{\mathbb{P}^{G,\succ}_{inv}\}$ *with prefix-subsumption succeeds, then* G *is unsatisfiable. (ii) If* G *is unsatisfiable, then every fair derivation* $\mathcal{S}$ *in* $\{\mathbb{P}^{G,\succ}_{inv}\}$ *with prefix-subsumption succeeds.*

PROOF. (i) is proved as in Theorem 7.17 (completeness of $\{\mathbb{P}^{G,\succ}_{inv}\}$). For (ii), we will use Theorem 7.17. Let us show that (ii) immediately follows from

(7.3) Let $\mathcal{T}$ be a fair derivation in $\{\mathbb{P}^{G,\succ}_{inv}\}$ and a path sequent Π occurs 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 7.17 $\mathcal{T}$ contains the path sequent ϵ. Then by (7.3), $\mathcal{S}$ must contain a path sequent that prefix-subsumes ϵ, but there is only one such path sequent: ϵ itself. So, it remains to prove (7.3).

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 item 6 of Lemma 7.18 guarantees 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:

(7.4) *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

(7.5) *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 (7.5) can be reformulated: every $\Pi \in S_i$ has a trace in $\mathcal{S}$.

Note that (7.3) obviously follows from a slightly more general statement

(7.6) *Every path sequent* Π *that appears in a fair derivation* $\mathcal{T}$ *in* $\{\mathbb{P}^{G,\succ}_{inv}\}$ *has a trace in* $\mathcal{S}$.

We will now prove (7.6), and then we are done. Let $\mathcal{T} = T_0 \triangleright T_1 \triangleright \ldots$. Take the least i such that $\Pi \in T_i$. (7.6) will be proved by induction on i. If $i = 0$, then Π is an axiom, therefore it belongs to S_0, and by (7.5) has a trace in $\mathcal{S}$. If $i > 0$, then Π is obtained by an inference I of $\mathbb{P}^{G,\succ}_{inv}$ 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 two statements.

(7.7) *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.*

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

(7.7) 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 $\pi_2 \sqsubseteq \pi'_2$. So if π_1 being in conflict with π_2 implies π'_1, π'_2 are conflicting, too. To prove (7.8), note the following: Π consists of prefixes of Π', and every path is greater than any its prefix with respect to $\succ$, so π is greater than any path sequent in Π.

Let us now return to the proof of (7.6).

1. 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:

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

(b) *Case* Γ *has the form* $\Gamma', \pi\wedge_l$, *where* Γ' *prefix-subsumes* Π. Consider the inference

$$\frac{\Gamma', \pi\wedge_l}{\Gamma', \pi}\ (\wedge_l). \tag{7.9}$$

Let us show that this inference is not redundant in $\mathbb{P}^{G,\succ}_{inv}$. Evidently, Γ', π contains no pair of paths of different modal lengths, since so is $\Gamma', \pi\wedge_l$. (7.7) guarantees that Γ', π contains no conflicting pairs of paths, because so is Π, π, and finally, (7.8) guarantees that $\pi \succ \Gamma'$, because $\pi \succ \Pi$.
Since inference (7.9) is nonredundant and $\mathcal{S}$ is fair, $\mathcal{S}$ contains this inference. By (7.5), $\mathcal{S}$ contains a trace of Γ', π; evidently, this trace is also a trace of Π, π.
Below we will no more prove that the inferences in $\mathcal{S}$ are nonredundant, since the proof is similar to the one in this case.

(c) *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 (7.5) 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.

2. The inference I has the form

$$\frac{\Pi_1, \pi\vee_l \quad \Pi_2, \pi\vee_r}{\Pi_1, \Pi_2, \pi}\ (\vee).$$

Take traces Γ_1 and Γ_2 of $\Pi_1, \pi\vee_l$ and $\Pi_2, \pi\vee_r$, respectively, in $\mathcal{S}$. Similar to the previous case, we consider the following subcases:

(a) *Case (either* Γ_1 *prefix-subsumes* Π_1 *or* Γ_2 *prefix-subsumes* Π_2*).* Then either Γ_1 or Γ_2 prefix-subsumes Π_1, Π_2, π.

(b) *Case* Γ_1 *contains* $\pi\vee_l$ *and* Γ_2 *contains* $\pi\vee_r$. Then Γ_1 has the form $\Gamma_1', \pi\vee_l$ and Γ_2 has the form $\Gamma_2', \pi\vee_r$ such that Γ_1 and Γ_2 prefix-subsume Π_1 and Π_2, respectively. Since $\mathcal{S}$ is fair, it contains an inference

$$\frac{\Gamma_1', \pi\vee_l \quad \Gamma_2', \pi\vee_r}{\Gamma_1', \Gamma_2', \pi}\ (\vee).$$

By (7.5), $\mathcal{S}$ contains a trace of $\Gamma_1', \Gamma_2', \pi$; evidently, this trace is also a trace of Π_1, Π_2, π.

(c) *Case (*Γ_1 *has the form* Γ_1', π', *where* Γ_1' *prefix-subsumes* Π_1 *and* $\pi' \sqsubset \pi\vee_l$*).* This case is similar to the respective case for the conjunction above.

(d) *Case (*Γ_2 *has the form* Γ_2', π', *where* Γ_2' *prefix-subsumes* Π_2 *and* $\pi' \sqsubset \pi\vee_r$*).* This case is symmetric to the previous one.

Other cases are considered in [Voronkov 2001].

The proof is complete. □

However, the completeness result for prefix-subsumption has some applications besides the possibility of removing subsumed sequents. Let us look at some 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 that 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 $\mathbb{P}^{G,\succ}_{inv}$ so that such paths do not occur in sequents at all. This modification is similar to the optimization used for the named calculus in Section 5.3. Such an optimization first appeared for classical logic in [Voronkov 1985], and then in a more accessible paper [Voronkov 1990*b*]. It has also been used in tableau calculi in [Degtyarev and Voronkov 1996*b*] as a *least conjunctive superformula* optimization. For logic **K** it has been implemented in the system KK [Voronkov 2000*a*].

Another interesting consequence of the use of prefix-subsumption is the following. Suppose that during search for a refutation we obtained a path sequent $\Delta = \Pi, \pi, \pi'$

$$\frac{}{\Gamma,\ A,\ \neg A}\ (Ax) \qquad \frac{\Gamma,\ A \quad \Gamma,\ B}{\Gamma,\ A \vee B}\ (\vee) \qquad \frac{\Gamma,\ A,\ B}{\Gamma,\ A \wedge B}\ (\wedge)$$

$$\frac{\Gamma,\ A(y)}{\Gamma,\ \exists x A(x)}\ (\exists) \qquad \frac{\Gamma,\ A(t)}{\Gamma,\ \forall x A(x)}\ (\forall)$$

The rule $(\exists)$ satisfies the eigenvariable condition: y is a variable not occurring in Γ.

Figure 30: Calculus $\mathbf{G}_{tab}$ for Grishin's logic

such that $\pi \sqsubseteq \pi'$. Consider the path sequent $\Gamma = \Pi, \pi', \pi'$. This sequent is prefix-subsumed by Δ, so it can be added to the search space without compromising soundness or completeness. But if we add Γ to the search space, by contraction we can derive Π, π' that subsumes Δ. Therefore Π, π, π' can be replaced by a simpler sequent Π, π', as soon as π is a prefix of π'.

8. Logics without the contraction rules

The use of the inverse method for proving the decidability of various logics has a long history. Gentzen [1934] proved the decidability of propositional intuitionistic logic essentially using the inverse method technique. Maslov [1966] used the inverse method to prove the decidability of several fragments of first-order classical logic.

In this section we demonstrate a nice application of the inverse method for the decidability of some ***contraction-free logics***: logics without the contraction rule. A number of such logics have been proposed. For example Grishin [1981] introduced such a logic with the purpose of providing alternative foundations for logic and set theory, while Ketonen and Weyhrauch [1984] introduced a contraction-free fragment of predicate calculus with the aim of defining "simply provable" formulas. Another example of such a logic is the linear fragment of linear logic [Girard, Lafont and Taylor 1989].

Let us show that the predicate version of Grishin's [1981] logic is decidable.[8] The decidability of this logic was proved by the inverse method in [Voronkov 1992]. The technique used in the proof is very general and can be used to prove the decidability of all other contraction-free logics mentioned here. The standard translation to negation normal form is also valid for Grishin's logic, so we present the logic for NNFs.

A sequent calculus for the predicate version of Grishin's logic, denoted $\mathbf{G}_{tab}$ is shown in Figure 30. Compare this calculus with the calculus $\mathbf{C}_{tab}$ for NNFs, see Figure 10 on page 205. The only difference between the two calculi is the rule $(\forall)$, in which the quantified formula $\forall x A(x)$ is not duplicated.

[8]Grishin writes that his logic is decidable but gives no proof of it.

The main aim of this section is to design an inverse calculus for Grishin's logic and prove the decidability of this logic using proof-theoretic properties of the calculus. With the technique developed so far, it is almost a mechanical exercise. For aesthetic reasons, we will show how to mix ingredients used so far in a slightly new way. For example, we use paths instead of names and make a first-order calculus that is technically simpler than those used previously by using a notion of variable sequence on a path.

The proof of decidability of Grishin's logic is based on the following ideas, developed in the rest this section

- Design a path calculus corresponding to the sequent calculus of $\mathbf{G}_{tab}$. Prove a Bisimulation Lemma for this calculi, thus obtaining completeness.
- Design a corresponding inverse path calculus. This time we will skip the construction of a ground inverse method calculus and build a free-variable calculus immediately. In the completeness proof for this calculus, we will merge the Subsequent Lemma and Lifting. Then we will use proof-theoretic properties of the inverse path calculus to prove the decidability.

8.1. Tableau path calculus and bisimulation lemma

It will be convenient for us to deal with occurrences of subformulas in the goal formula instead of subformulas themselves. We already know two ways of formalizing the notion of occurrence: name calculi and path calculi. Name calculi were introduced for predicate logics in Section 5.1, path calculi were introduced for propositional logic in Section 7. As one can see, path calculi are technically more convenient since paths contain a lot of information about the occurrence of a subformula. In this section we will extend path calculi to the first-order case, using the technique of designing first-order inverse method calculi.

8.1. DEFINITION *(path in the first-order case).* A ***path*** is any finite sequence of symbols $\wedge_l$, $\wedge_r$, $\vee_l$, $\vee_r$, $\forall$, $\exists$. Let G be a formula. The notion of ***path in*** G, and the ***subformula of*** G ***on a path*** π, are defined as in Definition 7.2 on page 234, but with the following additional cases.

Suppose that π is a G-path and $G|_\pi = F$. Then

- If F has the form $\forall x F_1$, then $\pi\forall$ is a G-path, and $G|_{\pi\forall} = F_1$. In this case we call π a $\forall$-***path***.
- If F has the form $\exists x F_1$, then $\pi\exists$ is a G-path, and $G|_{\pi\exists} = F_1$. In this case we call π a $\exists$-***path***.

In the propositional case, each occurrence of a subformula in the goal formula G can be uniquely identified by its path. In the first-order case, this is true for free subformulas, but not for arbitrary subformulas. Any subformula of G can be obtained from a free subformulas by a applying a substitution, so any subformula F of G can be identified by a pair π, σ such that $F = G|_\pi\sigma$. This would give us a formalization similar to those used for designing free-variable calculi for classical

$$\frac{}{\Gamma,\ \pi_1(\bar{s}),\ \pi_2(\bar{t})}\ (Ax)$$

$$\frac{\Gamma, \pi \vee_l (\bar{s}) \quad \Gamma,\ \pi \vee_r (\bar{s})}{\Gamma,\ \pi(\bar{s})}\ (\vee) \qquad \frac{\Gamma,\ \pi \wedge_l (\bar{s}),\ \pi \wedge_r (\bar{s})}{\Gamma,\ \pi(\bar{s})}\ (\wedge)$$

$$\frac{\Gamma,\ \pi\exists(\bar{s}, y)}{\Gamma,\ \pi(\bar{s})}\ (\exists) \qquad \frac{\Gamma,\ \pi\forall(\bar{s}, t)}{\Gamma,\ \pi(\bar{s})}\ (\forall)$$

The rule $(\exists)$ satisfies the eigenvariable condition: y is a variable not occurring in Γ or $\bar{s}$. In the rule (Ax) we have $G|_{\pi_1(\bar{s})} = \neg G|_{\pi_2(\bar{t})}$.

Figure 31: Tableau path calculus for Grishin's logic

logic in Section 3.3 and intuitionistic logic in Section 4.1. However, we do not want to work with such pairs and will try a simpler presentation in the spirit of a name calculus. We will use paths as names and use expressions like $\pi(t_1, \ldots, t_n)$, where π is a path, to encode arbitrary subformulas of the goal formula.

8.2. DEFINITION *(variable sequence and path name).* The notion of ***variable sequence on a G-path*** π, denoted $var(G, \pi)$, is defined as follows

1. $var(G, \epsilon)$ is the empty sequence of variables.
2. If $\pi\forall\!\!\!\exists_*$ is a path in G, then $var(G, \pi\forall\!\!\!\exists_*) = var(G, \pi)$.
3. If $var(G, \pi)$ is a sequence $(x_1, \ldots, x_n)$ and $G|_\pi$ has the form $\forall x F$ (respectively $\exists x F$), then $var(G, \pi\forall) = (x_1, \ldots, x_n, x)$ (respectively, $var(G, \pi\exists) = (x_1, \ldots, x_n, x)$).

Let π be a path in G, $var(G, \pi) = (x_1, \ldots, x_n)$ and $t_1, \ldots, t_n$ be a sequence of terms. Then we call the expression $\pi(t_1, \ldots, t_n)$ the ***G-path name*** of the formula $G|_\pi\{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$.

Consider, for example, the formula $G = \forall x(\exists y P(x, y) \vee \forall z(\neg P(x, z) \vee Q(z))$. Then $\pi = \forall \vee_r \forall \vee_r$ is a path in G, and we have $G|_\pi = Q(z)$ and $var(G, \pi) = (x, z)$. Further, $\forall \vee_r \forall \vee_r (f(a), b)$ is a G-path name of the subformula $Q(b)$. In the sequel we will simply write *path name*, since G will always be clear from the context. If $\pi(\bar{t})$ is a G-path name for a subformula F of G, we will use notation $G|_{\pi(\bar{t})}$ to denote F. Note that this notation is consistent with the one with used for propositional **K**.

It is easy to see that $var(G, \pi)$ contains all variables in $free(G|_\pi) \setminus free(G)$. In particular, if G is closed, then $var(G, \pi)$ contains all free variables of $G|_\pi$.

The ***tableau path calculus for Grishin's logic*** is shown in Figure 31. Its sequents are multisets of G-path names.

Similar to the case of logic **K**, we define the ***formula image*** of a path sequent $\pi_1(\bar{t}_1), \ldots, \pi_n(\bar{t}_n)$ as the sequent $G|_{\pi_1(\bar{t}_1)}, \ldots, G|_{\pi_1(\bar{t}_1)}$.

Using the same considerations as in Bisimulation Lemma 7.3 on page 235, we can prove

8.3. LEMMA (Bisimulation Lemma for Grishin's logic). *(i) Let D be a derivation in the tableau path calculus for G. Then the formula image of D is a derivation of G in $\mathbf{G}_{tab}$. (ii) Let D' be a derivation of a sequent $A_1, \ldots, A_n$ in $\mathbf{G}_{tab}$, and $\pi_1(\bar{t}_1), \ldots, \pi_n(\bar{t}_n)$ be G-path names of $A_1, \ldots, A_n$. Then there exists a derivation D of $\pi_1(\bar{t}_1), \ldots, \pi_n(\bar{t}_n)$ in the tableau path calculus for G such that D' is the formula image of D. (iii) Both (i) and (ii) also hold if "derivation" is replaced by "refutation" everywhere.* □

This lemma implies soundness and completeness of the tableau path calculus for G.

8.4. THEOREM (Completeness). *Let G be a closed formula. Then G has a refutation in $\mathbf{G}_{tab}$ if and only if ϵ has a refutation in the tableau path calculus for G.* □

8.2. Inverse path calculus and decidability

The construction of a free-variable inverse path calculus for Grishin's logic is now straightforward. We suggest the reader try constructing it before looking at Figure 32. There are two tricky rules: $\vee$ and $\wedge$. The rule $\vee$ in the tableau path calculus has a hidden contraction rule on Γ, which we should take care of in the inverse calculus. The rule $(\wedge)$ now gets split into three rules. We need three rules since we do not have the contraction rule any more, and the construction used in the proof of the Subsequent Lemma on page 190 may not work.

Subsequent Lemma and Lifting will now be merged using the following notion. We say that a path sequent Π_1 ***subsumes*** a path sequent Π_2 if there exists a substitution σ such that $\Pi_1\sigma$ is a submultiset of Π_2.

8.5. LEMMA (Subsumption Lemma). *Let Π_2 have a refutation in the tableau path calculus for Grishin's logic. Then there exists a path sequent Π_1 such that (i) Π_1 subsumes Π_2 and (ii) Π_1 has a refutation in the free-variable inverse path calculus for Grishin's logic.* □

As usual, we obtain soundness and completeness as a consequence of this lemma.

8.6. THEOREM (Completeness). *Let G be a closed formula. Then G has a refutation in $\mathbf{G}_{tab}$ if and only if ϵ has a refutation in the free-variable inverse path calculus for G.* □

We are just one step from proving the decidability of Grishin's logic. Let us note one useful property of the free-variable inverse path calculus. We call a path sequent ***prefix-free*** if for every two different occurrences of path names $\pi_1(\bar{s}), \pi_2(\bar{t})$ in Π, π_1 is not a prefix of π_2 and π_2 is not a prefix of π_1.

$$\frac{}{\Gamma,\ \pi_1(\bar{s}),\ \pi_2(\bar{t})}\ (Ax) \qquad \frac{\Gamma_1, \Delta_1, \pi \vee_l (\bar{s}) \quad \Gamma_2, \Delta_2,\ \pi \vee_r (\bar{t})}{(\Gamma_1,\ \Delta_1,\ \Delta_2,\ \pi(\bar{s}))\sigma}\ (\vee)$$

$$\frac{\Gamma,\ \pi \wedge_l (\bar{s})}{\Gamma,\ \pi(\bar{s})}\ (\wedge_l) \qquad \frac{\Gamma,\ \pi \wedge_r (\bar{t})}{\Gamma,\ \pi(\bar{t})}\ (\wedge_r) \qquad \frac{\Gamma,\ \pi \wedge_l (\bar{s}),\ \pi \wedge_r (\bar{t})}{(\Gamma,\ \pi(\bar{s}))\mathrm{mgu}(\bar{s}, \bar{t})}\ (\wedge)$$

$$\frac{\Gamma,\ \pi\exists(\bar{s}, y)}{\Gamma,\ \pi(\bar{s})}\ (\exists) \qquad \frac{\Gamma,\ \pi\forall(\bar{s}, t)}{\Gamma,\ \pi(\bar{s})}\ (\forall)$$

For the rule (Ax) the following conditions holds: there exists free subformulas $\neg A$ and B of G and a most general unifier σ of A and B such that $\pi_1(\bar{s})$ is a path name for $\neg A\sigma$ and $\pi_2(\bar{t})$ is a path name for $B\sigma$.
In the rule $(\vee)$ σ is a most general unifier of $\Gamma_1, \bar{s}$ and $\Gamma_2, \bar{t}$.
The rule $(\exists)$ satisfies the eigenvariable condition: y is a variable not occurring in Γ or $\bar{s}$.

Figure 32: Free-variable inverse path calculus for Grishin's logic

8.7. Lemma. *Let D be a refutation of ϵ in the free-variable inverse path calculus and Π be a path sequent in D. Then Π is prefix-free.*

Proof. Suppose, by contradiction, that some path sequent Π in D is not prefix-free. By inspecting the rules of the free-variable inverse path calculus one can see that any path sequent below Π in D is also not prefix-free. Therefore, the bottom sequent of the derivation must contain at least two path names, so it cannot be ϵ. □

The decidability of Grishin's logic will be proved as follows. We will build a partial order $>$ on path sequents which satisfies the following properties:

1. for every inference in the free-variable path calculus the conclusion of the inference is strictly greater than each premise w.r.t. $>$;
2. in every infinite increasing w.r.t. $>$ chains of path sequents, there is a path sequent that is not prefix-free, and therefore cannot occur in a refutation of ϵ.

These properties will immediately imply the decidability of Grishin's logic.

First, we construct such a partial order $>$ for multisets of paths. For two multisets of paths Π_1 and Π_2 we have $\Pi_1 > \Pi_2$ if Π_1 can be obtained from Π_2 by a finite sequence of the following steps:

- addition of any path;
- replacement of any path by its prefix.

For example, we have $\forall, \exists\wedge_l > \forall\vee_r$ because we can obtain $\forall, \exists\wedge_l$ from $\forall\vee_r$ in two steps:

- addition of the path $\exists\wedge_l$;

- replacement of the path $\forall\vee_r$ by its prefix $\forall$.

It is not hard to argue that $>$ is a partial order.

For path sequents we define $\pi_1(\bar{s}_1), \ldots, \pi_n(\bar{s}_n) > \pi'_1(\bar{t}_1), \ldots, \pi'_m(\bar{t}_n)$ if and only if $\pi_1, \ldots, \pi_n > \pi'_1, \ldots, \pi'_m$.

Let us prove the above two properties. When we prove them, we will freely shift from path names $\pi(\bar{t})$ to paths π and back.

1. *For every inference in the free-variable path calculus the conclusion of the inference is strictly greater than each premise w.r.t. $>$.* This property trivially holds for every inference of the calculus. For example, in the rule $(\vee)$ the conclusion can be obtained from the left premise by adding all paths in Δ_2 and replacement of $\pi\vee_l$ by its prefix π.
2. *In every infinite increasing w.r.t. $>$ chains of path sequents, there is a path sequent that is not prefix-free, and therefore cannot occur in a refutation of ϵ.* Suppose that we have an infinite increasing chain of paths sequents

$$\ldots \Pi_2 > \Pi_1 > \Pi_0.$$

Note that for any path sequent Π_i, we can replace a path by its prefix only a finite number of times. This means that the number of path names in Π_i's is not bound by any finite number. Since the number of G-paths is finite, this means that some Π_i (with a large enough number of elements) contains two different occurrences of the same path.

Note that this observation can be generalized to

8.8. LEMMA. *The depth of derivations of ϵ is bound by the number $d(G) = m \cdot n$, where m is the length of the longest G-path and n is the number of all G-paths.*

8.9. THEOREM. *Grishin's logic is decidable.*

PROOF. By Completeness Theorem 8.6, it is enough to show how to decide whether ϵ has a refutation in the free-variable inverse path calculus for G. By the above observation, the depth of any refutation of ϵ is bounded by a finite number $d(G)$ that can be effectively computed for G. But we know that the free-variable inverse path calculus has the finite rule property, so we can effective generate all derivations up to this depth. □

9. Conclusion

9.1. History

In this section be briefly overview the history of the inverse method. We do not mention *all* publications on the inverse method here, we also do not mention who was the first to treat a particular nonclassical logic. Rather, we only consider papers that introduce some novelties in the technique, for example free-variable calculi.

The first paper that discussed the inverse method was [Gentzen 1934]. Gentzen proposed a decision procedure for intuitionistic propositional logic, as an application of cut-free sequent calculi. Let us quote the relevant part of the paper.

> On the basis of the *Hauptsatz* we can state a simple procedure for deciding of a formula of propositional logic — i.e., a formula without object variables — whether or not it is classically or intuitionistically true. (For classical propositional logic a simple solution has actually been known for some time, cf., e.g. p.11 of Hilbert-Ackermann.)
>
> First we prove the following *lemma*:
>
> A sequent in whose antecedent one and the same formula does not occur more then three times as an S-formula, and in whose succedent, furthermore, one and the same formula does not occur more then three times as an S-formula, will be called a "reduced sequent". The following lemma now holds:
>
> Every LJ- or LK-derivation whose endsequent is reduced, may be transformed into an LJ- or LK-derivation with the same endsequent, in which all sequents are reduced (and in which no cuts occur if the original derivation did not contain any).
>
> Consider now a sequent not containing an object variable. We wish to decide whether or not it is intuitionistically or classically true. We can begin by taking in its place an equivalent reduced sequent $\mathfrak{S}\mathfrak{q}$.
>
> The number of all reduced sequents whose S-formulae are subformulae of the S-formulae of $\mathfrak{S}\mathfrak{q}$ is obviously finite. The decision procedure may therefore be carried out without further complications in the following way:
>
> We consider the finite system of sequents in question and investigate first of all, which of these sequents are basic sequents[9]. Then we examine each of the remaining sequents to determine whether there occurs an inference figure in which the sequent in question is the lower sequent and in which there occur as upper sequents one or two of those sequents that have already been found to be derivable. If this is the case, the sequent is added to the derivable sequents. (All this is obviously decidable.) We continue in this way until either the sequent $\mathfrak{S}\mathfrak{q}$ itself turns out to be derivable, or until the procedure yields no new derivable sequents. In the latter case the sequent $\mathfrak{S}\mathfrak{q}$ is not derivable at all in the calculus under consideration (LJ or LK). We have therefore succeeded in establishing the validity of that sequent.
>
> Our decision procedure can be formulated in a way better suited to the needs of practical application; yet the above presentation was intended only to indicate a possibility in principle.

The reason for the introduction of "reduced sequents" was due to the fact that in Gentzen's original system sequents were sequences of formulas and he had several structure rules for handling such sequents.

The term "inverse method" and an inverse method calculus for classical logic were introduced in Maslov [1964]. The calculus was defined for classical first-order logic without equality and function symbols, however it was noted that

> "it is a relatively simple matter to develop the inverse method to include the case of the classical predicate calculus with constant functional symbols"

[9] i.e. axioms.

Skolemization was not used, and the method was applied to formulas of the form

$$Q_1x_1 \dots Q_nx_n(\bigvee_{i=1}^{\mu} \bigwedge_{j=1}^{\delta_i} D_{i,j}).$$

where $Q_1x_1 \dots Q_nx_n$ is a quantifier prefix and $D_{i,j}$ are disjunctions of literals. Maslov [1964] did not refer to sequent calculi and the subformula property, and the method was presented as based on Herbrand's theorem. It was emphasized as well that the notion of metavariables (corresponding to free variables in the modern terminology) proposed by Shanin in 1962 served as the author's point of departure.

Maslov [1964] showed how to apply the method for establishing the decidability of classes of formulas, in particular the class of formulas of the form

$$\forall^*\exists^*\forall^*(\bigvee_{i=1}^{\mu} \bigwedge_{j=1}^{\delta_i} D_{i,j}).$$

where $\delta_1 \leq 2, \dots \delta_\mu \leq 2$ (called the Maslov class in [Börger, Grädel and Gurevich 1997]).

Maslov [1966] applied the inverse method to prove in a uniform way the *decidability of several classes of formulas* of the first-order classical predicate calculus, including some new classes. Maslov [1967] modified the inverse method to *arbitrary formulas with function symbols*. Relationships between the *inverse method and (hyper)resolution* were noted by [Maslov 1969, Maslov 1971*b*, Kuechner 1971]. The first implementation of the inverse method for classical logic is due to Davydov, Maslov, Mints, Orevkov and Slissenko [1969]. Maslov [1971*a*] showed how to generalize the inverse method to *logic with equality*, but not in the form of a free-variable calculus[10]. Mints [1981] (published in Russian) was the first to apply the inverse method (under the name of resolution) to *nonclassical* propositional logics. This and other early papers on the inverse method for nonclassical logics are overviewed in [Mints 1990]. A *free-variable* inverse calculus for classical logic without equality was introduced in [Voronkov 1985], a more accessible reference is [Voronkov 1990*b*]. Inverse calculi for *nonclassical predicate logics appeared* in [Voronkov 1992], already with free variables. This paper also proposed a uniform technique for proving redundancy criteria. A *free-variable method* for classical logic *with equality* was introduced in [Degtyarev and Voronkov 1995*a*] with a number of redundancy criteria, for example those based on reduction orderings and the basic strategy[11]. A first implementation of the inverse method for predicate intuitionistic logic is described in [Tammet 1996]. *Path calculi* were introduced in [Voronkov 2000*b*].

[10]Maslov [1971*a*] formulated completeness of the inverse method in terms of existence of a substitution with some properties, so his formulation were halfway to free-variable methods, since no algorithm for finding such a substitution has been proposed. In fact, the problem of the existence of such a substitution for logic with equality is undecidable, since one can reduce to it simultaneous rigid E-unification [Degtyarev and Voronkov 1996*f*].

[11]Free-variable methods for some nonclassical logics with equality do not exist, in a certain sense, for details see [Voronkov 1998*c*].

9.2. *Limits of the technique and future research*

9.2.1. *More expressive description logics*

It seems that the inverse method is applicable to logics having a suitable subformula property, including logics with the analytic cut rule. However, there is a long way to go in applying it to more expressive logics, for example, some description logics covered in [Calvanese, Giacomo, Lenzerini and Nardi 2001] (Chapter 23 of this Handbook). Such logics are usually defined by a possible world semantics or by tableau systems with explicit notation for worlds. It is unclear whether such tableau systems can be easily made into inverse calculi: this requires further research.

9.2.2. *New completeness proofs*

Moreover, proving completeness of inverse calculi using the detour through tableau calculi may be unnecessary, provided that new methods to prove completeness are developed. Development of such methods is a very interesting research topic. If such methods are developed, it is possible that completeness proofs for calculi with redundancy criteria will be considerably simplified.

9.2.3. *Implementation*

It would be interesting to implement the inverse method for nonclassical propositional logics efficiently, so that it can compete with the best available systems, for example FaCT, DLP [Horrocks and Patel-Schneider 1999] or ∗SAT [Giunchiglia, Giunchiglia and Tacchella 2000]. The current implementation of the system Kꓘ based on the inverse method [Voronkov 2000*a*] is still not as fast as these systems.

For first-order intuitionistic logic the inverse method was implemented efficiently by Tammet [1996], but the implementations of first-order nonclassical logics are not so advanced as the implementations of propositional logics, and especially description logics.

9.2.4. *Bidirectonal systems*

It would be interesting to develop bidirectional theorem proving systems, where proof-search will be carried out by a combination of top-down (tableaux) and bottom-up (inverse) proof-search methods. This requires new methods of proving completeness of calculi with redundancy criteria, in order to ensure that the criteria developed for the tableau part and the inverse part are compatible.

9.3. *Warning*

If you had enough patience to read this chapter to the very end, you become very experienced in cooking. You can take almost any logic (well, any logic complying with the Subformula Property), and make a meal out of it. The aim of this subsection is to warn you (see also the Fast Food Warning in Section 5.7): with prolific food you will become fat and your friends and colleagues will despise you[12]. The calculi for the inverse method can be obtained in an essentially automatic way for most logics with the subformula property. To obtain *more elegant* calculi by using variations on the initial calculi or by using the completeness proofs directly is an appealing task. To obtain inverse method calculi with *new unusual features* is of a definite interest. To find *new redundancy criteria* and prove a combination of them complete is a very creative task. To *implement* the inverse method for propositional nonclassical logics *with efficiency comparable with that of the tableau method* is a challenge. To design a *bidirectional system with redundancy criteria* is one of the most important problems in this area. The design and implementation of such systems can become the *major research area for logics used in applications*, for example description logics.

Acknowledgments

We thank the second readers whose remarks have improved the article considerably. However, the authors are the only responsible for all inaccuracies that can be present.[13]

Bibliography

Andrews P. [2001], Classical type theory, *in* A. Robinson and A. Voronkov, eds, 'Handbook of Automated Reasoning', Vol. II, Elsevier Science, chapter 15, pp. 965–1007.

Apt K. [1990], Logic programming, *in* J. Van Leeuwen, ed., 'Handbook of Theoretical Computer Science', Vol. B: Formal Methods and Semantics, Elsevier Science, Amsterdam, chapter 10, pp. 493–574.

Auffray I., Enjalbert P. and Hebrard J.-J. [1990], 'Strategies for modal resolution: Results and problems', *Journal of Automated Reasoning* **6**, 1–38.

Baader F. and Snyder W. [2001], Unification theory, *in* A. Robinson and A. Voronkov, eds, 'Handbook of Automated Reasoning', Vol. I, Elsevier Science, chapter 8, pp. 445–532.

Baaz M., Egly U. and Leitsch A. [2001], Normal form transformations, *in* A. Robinson and A. Voronkov, eds, 'Handbook of Automated Reasoning', Vol. I, Elsevier Science, chapter 5, pp. 273–333.

Baaz M., Fermüller C. and Salzer G. [2001], Automated deduction for many-valued logics, *in* A. Robinson and A. Voronkov, eds, 'Handbook of Automated Reasoning', Vol. II, Elsevier Science, chapter 20, pp. 1355–1402.

Bachmair L. and Ganzinger H. [2001], Resolution theorem proving, *in* A. Robinson and A. Voronkov, eds, 'Handbook of Automated Reasoning', Vol. I, Elsevier Science, chapter 2, pp. 19–99.

[12]There was a lot of activity in Russia in the area of Lyapunov's vector functions method. In the 1970s Valeri Matrosov with his group implemented a system that proved theorems in this area in a completely automatic way. It has been shown that many articles published in respected scientific journals could have been written by a computer. Yuri Ershov characterized the main achievement of Matrosov thus: they proved that a part of mathematics has no right to exist. It is definitely *not* the aim of this chapter to seize the research on the inverse method.

[13]We would like to mention that the second second reader has a different viewpoint on the history of the inverse method, see e.g., [Maslov, Mints and Orevkov 1983].

BARENDREGT H. AND GEUVERS H. [2001], Proof-assistants using dependent type systems, *in* A. Robinson and A. Voronkov, eds, 'Handbook of Automated Reasoning', Vol. II, Elsevier Science, chapter 18, pp. 1149–1238.

BETH E. W. [1956], 'Semantic construction of intuitionistic logic', *Noord-Hollandsche Uitgevers Maatschappij* .

BOCKMAYR A. AND WEISPFENNING V. [2001], Solving numerical constraints, *in* A. Robinson and A. Voronkov, eds, 'Handbook of Automated Reasoning', Vol. I, Elsevier Science, chapter 12, pp. 749–842.

BÖRGER E., GRÄDEL E. AND GUREVICH Y. [1997], *The Classical Decision Problem*, Springer Verlag.

BOWEN K. [1979], *Model Theory for Modal Logic*, Vol. 127 of *Synthese Library*, D. Reidel.

BOY DE LA TOUR T. [1990], Minimizing the number of clauses by renaming, *in* M. Stickel, ed., 'Proc. 10th CADE', Vol. 449 of *Lecture Notes in Artificial Intelligence*, pp. 558–572.

BUNDY A. [2001], The automation of proof by mathematical induction, *in* A. Robinson and A. Voronkov, eds, 'Handbook of Automated Reasoning', Vol. I, Elsevier Science, chapter 13, pp. 845–911.

CALVANESE D., GIACOMO G. D., LENZERINI M. AND NARDI D. [2001], Reasoning in expressive description logics, *in* A. Robinson and A. Voronkov, eds, 'Handbook of Automated Reasoning', Vol. II, Elsevier Science, chapter 23, pp. 1581–1634.

CHAGROV A. AND ZAKHARYASCHEV M. [1996], *Modal Logic*, Oxford University Press.

CHOU S. C. AND GAO X. S. [2001], Automated reasoning in geometry, *in* A. Robinson and A. Voronkov, eds, 'Handbook of Automated Reasoning', Vol. I, Elsevier Science, chapter 11, pp. 705–747.

CLARKE E. AND SCHLINGLOFF H. [2001], Model checking, *in* A. Robinson and A. Voronkov, eds, 'Handbook of Automated Reasoning', Vol. II, Elsevier Science, chapter 24, pp. 1635–1790.

COMON H. [2001], Inductionless induction, *in* A. Robinson and A. Voronkov, eds, 'Handbook of Automated Reasoning', Vol. I, Elsevier Science, chapter 14, pp. 913–962.

CURRY H. [1963], *Foundations of Mathematical Logic*, McGraw-Hill, New York.

DAVIS M. [2001], The early history of automated deduction, *in* A. Robinson and A. Voronkov, eds, 'Handbook of Automated Reasoning', Vol. I, Elsevier Science, chapter 1, pp. 3–15.

DAVYDOV V., MASLOV S., MINTS G., OREVKOV V. AND SLISSENKO A. [1969], 'A computer algorithm of establisihing deducibility based on the inverse method', *Zapiski Nauchnyh Seminarov LOMI* **16**, 8–19.

DE NIVELLE H., SCHMIDT R. AND HUSTADT U. [2000], 'Resolution-based methods for modal logics', *Logic Journal of the IGPL* **8**(3), 265–292.

DEGTYAREV A. AND VORONKOV A. [1994*a*], Equality elimination for semantic tableaux, UPMAIL Technical Report 90, Uppsala University, Computing Science Department.

DEGTYAREV A. AND VORONKOV A. [1994*b*], Equality elimination for the inverse method and extension procedures, UPMAIL Technical Report 92, Uppsala University, Computing Science Department.

DEGTYAREV A. AND VORONKOV A. [1995], Equality elimination for the inverse method and extension procedures, *in* C. Mellish, ed., 'Proc. International Joint Conference on Artificial Intelligence (IJCAI)', Vol. 1, Montréal, pp. 342–347.

DEGTYAREV A. AND VORONKOV A. [1996*a*], 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.

DEGTYAREV A. AND VORONKOV A. [1996*b*], 'The undecidability of simultaneous rigid E-unification', *Theoretical Computer Science* **166**(1–2), 291–300.

DEGTYAREV A. AND VORONKOV A. [1996*c*], What you always wanted to know about rigid E-unification, *in* J. Alferes, L. Pereira and E. Orlowska, eds, 'Logics in Artificial Intelligence.

European Workshop, JELIA'96', Vol. 1126 of *Lecture Notes in Artificial Intelligence*, Évora, Portugal, pp. 50–69.

DEGTYAREV A. AND VORONKOV A. [2001*a*], Equality reasoning in sequent-based calculi, *in* A. Robinson and A. Voronkov, eds, 'Handbook of Automated Reasoning', Vol. I, Elsevier Science, chapter 10, pp. 609–704.

DEGTYAREV A. AND VORONKOV A. [2001*b*], The inverse method, *in* A. Robinson and A. Voronkov, eds, 'Handbook of Automated Reasoning', Vol. I, Elsevier Science, chapter 4, pp. 179–272.

DERSHOWITZ N. AND PLAISTED D. [2001], Rewriting, *in* A. Robinson and A. Voronkov, eds, 'Handbook of Automated Reasoning', Vol. I, Elsevier Science, chapter 9, pp. 533–608.

DIX J., FURBACH U. AND NIEMELÄ I. [2001], Nonmonotonic reasoning: Towards efficient calculi and implementations, *in* A. Robinson and A. Voronkov, eds, 'Handbook of Automated Reasoning', Vol. II, Elsevier Science, chapter 19, pp. 1241–1354.

DOWEK G. [2001], Higher-order unification and matching, *in* A. Robinson and A. Voronkov, eds, 'Handbook of Automated Reasoning', Vol. II, Elsevier Science, chapter 16, pp. 1009–1062.

DYCKHOFF R. [1992], 'Contraction-free sequent calculi for intuitionistic logic', *Journal of Symbolic Logic* **57**(3), 795–807.

FERMÜLLER C., LEITSCH A., HUSTADT U. AND TAMMET T. [2001], Resolution decision procedures, *in* A. Robinson and A. Voronkov, eds, 'Handbook of Automated Reasoning', Vol. II, Elsevier Science, chapter 25, pp. 1791–1849.

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

FITTING M. [1996], *First Order Logic and Automated Theorem Proving*, Springer Verlag, New York. Second edition.

FITTING M. AND MENDELSOHN R. [1998], *First-Order Modal Logic*, Kluwer Academic Publishers.

FITTING M., THALMANN L. AND VORONKOV A. [2000], Term-modal logics, *in* R. Dyckhoff, ed., 'Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX 2000)', Vol. 1847 of *Lecture Notes in Artificial Intelligence*, Springer Verlag, pp. 220–236.

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. Reprinted from [Gentzen 1934].

GIRARD J.-Y., LAFONT Y. AND TAYLOR P. [1989], *Proofs and types*, Cambridge University Press.

GIUNCHIGLIA E., GIUNCHIGLIA F. AND TACCHELLA A. [2000], The SAT-based approach for classical modal logics, *in* E. Lamma and P.Mello, eds, 'AI*IA 99: Advanced in Artificial Intelligence. 6th Congress of the Italian Association for Artificial Intelligence', Vol. 1792 of *Lecture Notes in Artificial Intelligence*, Springer Verlag, Bologna, Italy, pp. 95–106.

GRISHIN V. [1981], 'Predicate and set-theoretic calculi based on logic without the contraction rule (in Russian)', *Izvestiya Akad. Nauk SSSR* **45**(1), 47–68.

HÄHNLE R. [2001], Tableaux and related methods, *in* A. Robinson and A. Voronkov, eds, 'Handbook of Automated Reasoning', Vol. I, Elsevier Science, chapter 3, pp. 100–178.

HINTIKKA K. [1955], 'Form and content in quantification theory', *Acta Philosophica Fennica* **8**, 7–55.

HORROCKS I. AND PATEL-SCHNEIDER P. [1999], 'Optimising description logic subsumption', *Journal of Logic and Computation* **9**(3), 267–293.

HUDELMAIER J. [1993], 'An $O(n \log n)$-space decision procedure for intuitionistic propositional logic', *Journal of Logic and Computation* **3**(1), 63–75.

KANGER S. [1957], *Provability in Logic*, Vol. 1 of *Studies in Philosophy*, Almqvist and Wicksell, Stockholm.

KANGER S. [1963], A simplified proof method for elementary logic, *in* P. Braffort and D. Hirschberg, eds, 'Computer Programming and Formal Systems', North Holland, pp. 87–94. Reprinted as [Kanger 1983].

KANGER S. [1983], A simplified proof method for elementary logic, *in* J. Siekmann and G. Wrightson, eds, 'Automation of Reasoning. Classical Papers on Computational Logic', Vol. 1, Springer Verlag, pp. 364–371. Reprinted from [Kanger 1963].

KETONEN J. AND WEYHRAUCH R. [1984], 'A decidable fragment of predicate calculus', *Theoretical Computer Science* **32**(3), 297–309.

KLEENE S. [1952], 'Permutability of inferences in Gentzen's calculi LK and LJ', *Memoirs of the American Mathematical Society* **10**, 11–18.

KLEENE S. [1967], *Mathematical Logic*, John Wiley and Sons.

KOWALSKI R. AND KUEHNER D. [1971], 'Linear resolution with selection function', *Artificial Intelligence* **2**, 227–260.

KRIPKE S. [1963], 'Semantical considerations on modal logics', *Acta Philosophica Fennica, Modal and Many-Valued Logics* pp. 83–94.

KUECHNER D. [1971], A note on the relation between resolution and Maslov's inverse method, *in* B. Meltzer and D. Michie, eds, 'Machine Intelligence', Vol. 6, American Elsevier, pp. 73–76.

LETZ R. AND STENZ G. [2001], Model elimination and connection tableau procedures, *in* A. Robinson and A. Voronkov, eds, 'Handbook of Automated Reasoning', Vol. II, Elsevier Science, chapter 28, pp. 2015–2114.

LIFSCHITZ V. [1989], 'What is the inverse method?', *Journal of Automated Reasoning* **5**(1), 1–23.

MAEHARA S. [1954], 'Eine Darstellung der intuitionistischen Logik in der klassischen', *Nagoya Math. J.* **1954**, 45–64.

MASLOV S. [1964], 'The inverse method of establishing deducibility in the classical predicate calculus', *Soviet Mathematical Doklady* **5**, 1420–1424.

MASLOV S. [1966], 'An application of the inverse method to the theory of decidable fragments of the classical calculus', *Soviet Mathematical Doklady* **159**(1), 17–20.

MASLOV S. [1967], 'An inverse method for establishing deducibility of nonprenex formulas of the predicate calculus', *Soviet Mathematical Doklady* **172**(1), 22–25. Reprinted as [Maslov 1983*a*].

MASLOV S. [1968], The inverse method of establishing deducibility of logical calculi (in Russian), *in* 'Collected Works of MIAN', Vol. 98, Nauka, Moscow, pp. 26–87.

MASLOV S. [1969], 'Relationship between tactics of the inverse method and the resolution method (in Russian)', *Zapiski Nauchnyh Seminarov LOMI* **16**. Reprinted as [Maslov 1983*c*].

MASLOV S. [1971*a*], 'The generalization of the inverse method to predicate calculus with equality (in Russian)', *Zapiski Nauchnyh Seminarov LOMI* **20**, 80–96. English translation in: Journal of Soviet Mathematics 1, no. 1.

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

MASLOV S. [1983*a*], An inverse method for establishing deducibility of nonprenex formulas of the predicate calculus, *in* J.Siekmann and G.Wrightson, eds, 'Automation of Reasoning (Classical papers on Computational Logic)', Vol. 2, Springer Verlag, pp. 48–54. Reprinted from [Maslov 1967].

MASLOV S. [1983*b*], On strategies of resolution and of the inverse method, *in* M. G., ed., 'Mathematical Logic and Automatic Theorem Proving', Nauka, Moscow, pp. 327–332.

MASLOV S. [1983*c*], Relationship between tactics of the inverse method and the resolution method, *in* J.Siekmann and G.Wrightson, eds, 'Automation of Reasoning (Classical papers on Computational Logic)', Vol. 2, Springer Verlag, pp. 264–272. Reprinted from [Maslov 1969].

MASLOV S. AND MINTS G. [1983], The proof-search theory and the inverse method (in Russian), *in* M. G., ed., 'Mathematical Logic and Automatic Theorem Proving', Nauka, Moscow, pp. 291–314.

MASLOV S., MINTS G. AND OREVKOV V. [1983], Mechanical proof-search and the theory of logical deduction in the USSR, *in* J.Siekmann and G.Wrightson, eds, 'Automation of Reasoning (Classical papers on Computational Logic)', Vol. 1, Springer Verlag, pp. 29–38.

MINTS G. [1981], Resolution calculi for the non-classical logics (in Russian), *in* '9th Soviet Symp. on Cybernetics', Vol. 2, VINITI, Moscow, pp. 34–36.

MINTS G. [1990], Gentzen-type systems and resolution rules. Part I. Propositional logic, *in* P. Martin-Löf and G. Mints, eds, 'COLOG-88', Vol. 417 of *Lecture Notes in Computer Science*, Springer Verlag, pp. 198–231.

MINTS G. [1993*a*], Gentzen-type systems and resolution rule, Part II: Predicate logic, *in* J. Oikkonen and J. Väänänen, eds, 'Proc. ASL Summer Meeting, Logic Colloquium'90', Vol. 2 of *Lecture Notes in Logic*, Springer Verlag, Helsinki, Finland, 15–22 July 1990, pp. 163–190.

MINTS G. [1993*b*], '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.

NIEUWENHUIS R. AND RUBIO A. [2001], Paramodulation-based theorem proving, *in* A. Robinson and A. Voronkov, eds, 'Handbook of Automated Reasoning', Vol. I, Elsevier Science, chapter 7, pp. 371–443.

NONNENGART A. AND WEIDENBACH C. [1998], On generating small clause normal forms, *in* C. Kirchner and H. Kirchner, eds, 'Automated Deduction — CADE-15. 15th International Conference on Automated Deduction', Vol. 1421 of *Lecture Notes in Artificial Intelligence*, Springer Verlag, Lindau, Germany, pp. 397–411.

NONNENGART A. AND WEIDENBACH C. [2001], Computing small clause normal forms, *in* A. Robinson and A. Voronkov, eds, 'Handbook of Automated Reasoning', Vol. I, Elsevier Science, chapter 6, pp. 335–367.

OHLBACH H., NONNENGART A., DE RIJKE M. AND GABBAY D. [2001], Encoding two-valued nonclassical logics in classical logic, *in* A. Robinson and A. Voronkov, eds, 'Handbook of Automated Reasoning', Vol. II, Elsevier Science, chapter 21, pp. 1403–1486.

OREVKOV V. [1983], The British museum algorithm may be more efficient than resolution, *in* M. G., ed., 'Mathematical Logic and Automatic Theorem Proving', Nauka, Moscow, pp. 314–326.

PAULSON L. [1990], Isabelle: The next 700 theorem provers, *in* P. Odifreddi, ed., 'Logic and Computer Science', Academic Press, pp. 361–386.
URL: *http://www.cl.cam.ac.uk/Research/Reports/TR143-lcp-experience.dvi.gz*

PFENNING F. [2001], Logical frameworks, *in* A. Robinson and A. Voronkov, eds, 'Handbook of Automated Reasoning', Vol. II, Elsevier Science, chapter 17, pp. 1063–1147.

PLIUŠKEVIČIUS R. [1965], 'On a variant of the constructive predicate calculus without structural deduction rules', *Soviet Mathematical Doklady* **6**, 416–419.

RAMAKRISHNAN I., SEKAR R. AND VORONKOV A. [2001], Term indexing, *in* A. Robinson and A. Voronkov, eds, 'Handbook of Automated Reasoning', Vol. II, Elsevier Science, chapter 26, pp. 1853–1964.

SHANKAR N. [1992], Proof search in the intuitionistic sequent calculus, *in* D. Kapur, ed., '11th International Conference on Automated Deduction', Vol. 607 of *Lecture Notes in Artificial Intelligence*, Springer Verlag, Saratoga Springs, NY, USA, pp. 522–536.

SMULLYAN R. [1968], *First-Order Logic*, Springer Verlag.

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, 'Automated Deduction — CADE-13', Vol. 1104 of *Lecture Notes in Computer Science*, New Brunswick, NJ, USA, pp. 2–16.

TAMMET T. [1997], Resolution, inverse method and the sequent calculus, *in* G. Gottlob, A. Leitsch and D. Mundici, eds, 'Computational Logic and Proof Theory. 5th Kurt Gödel Colloquium, KGC'97', Vol. 1289 of *Lecture Notes in Computer Science*, Vienna, Austria, pp. 65–83.

TROELSTRA A. S. AND SCHWICHTENBERG H. [1996], *Basic Proof Theory*, Vol. 43 of *Cambridge Tracts in Theoretical Computer Science*, Cambridge University Press.

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*, Springer Verlag, Saratoga Springs, NY, USA, pp. 648–662.

VORONKOV A. [1998], 'Proof-search in intuitionistic logic with equality, or back to simultaneous rigid E-unification', *Journal of Automated Reasoning* **21**, 205–231.

VORONKOV A. [1999], KK: a theorem prover for K, *in* H. Ganzinger, ed., 'Automated Deduction—CADE-16. 16th International Conference on Automated Deduction', Lecture Notes in Artificial Intelligence, Trento, Italy, pp. 383–387.

VORONKOV A. [2000*a*], How to decide K using ꓘ, *in* A. Cohn, F. Giunchiglia and B. Selman, eds, 'Principles of Knowledge Representation and Reasoning (KR'2000)', pp. 198–209.

VORONKOV A. [2000*b*], How to optimize proof-search in modal logics: a new way of proving redundancy criteria for sequent calculi, *in* 'LICS'2000'.

VORONKOV A. [2001], 'How to optimize proof-search in modal logics: new methods of proving redundancy criteria for sequent calculi', *ACM Transactions on Computational Logic* . To appear.

WAALER A. [2001], Connections in nonclassical logics, *in* A. Robinson and A. Voronkov, eds, 'Handbook of Automated Reasoning', Vol. II, Elsevier Science, chapter 22, pp. 1487–1578.

WEIDENBACH C. [2001], Combining superposition, sorts and splitting, *in* A. Robinson and A. Voronkov, eds, 'Handbook of Automated Reasoning', Vol. II, Elsevier Science, chapter 27, pp. 1965–2013.

WEIDENBACH C., GAEDE B. AND ROCK G. [1996], SPASS & FLOTTER. Version 0.42, *in* M. McRobbie and J. Slaney, eds, 'Automated Deduction — CADE-13', Vol. 1104 of *Lecture Notes in Computer Science*, New Brunswick, NJ, USA, pp. 141–145.

ZAMOV N. [1989], 'Maslov's inverse method and decidable classes', *Annals of Pure and Applied Logic* **42**(2), 165–194.

Index

Symbols

$E(t_1, \ldots, t_n)$ **186**
$E\theta$ — result of application of substitution θ to expression E **186**
G-path **234**
$\Box$-path **234**
$\Box\Gamma$ **217**
$\Gamma \dot{\subseteq} \Delta$ — Γ is submultiset of Δ **185**
✡ **239**
$\Pi\Box$ **235**
$\alpha \cdot \theta$ **195**
⩙ **239**
$⩙_*$ **239**
$\cup$ — union of substitutions **186**
$\Diamond$-path **234**
ϵ **234**
$\exists$-path **256**
$\forall$-path **256**
$\dot{\in}$ — multiset membership **185**
$\dot{-}$ — multiset difference **185**
$\succ^{mul}$ — multiset extension of $\succ$ **248**
$\overline{L}$ — literal complementary to L **203**
$\pi \succ \Gamma$ **242**
$\rightleftharpoons$ — equal by definition **186**
$\sigma \leq \theta$— σ is more general than δ **196**
$\sigma\theta$ — composition of σ, θ **186**
σ_{-x} — substitution obtained by redefinition of σ in x **196**
$\sigma\restriction_V$ — restriction of substitution σ to the set of variables V **196**
$G|_{\pi(\bar{t})}$ **257**
$\succeq$ **240**
$\triangleright$ **249**
$\vee$-fork **237**
$\vee$-path **234**
$\vee_*$ **239**
$\wedge$-path **234**
$\wedge_*$ **239**
ξ — goal signed formula **184**
$\{\mathbb{P}^{G,\succ}_{inv}\}$ **250**
$\{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$ — notation for substitutions **185**

A

admissible substitution **186**
Ax_G **224**

B

brothers **239**

C

$\mathbf{C}^G_{inv}$ — free-variable inverse calculus for formulas in NNF 205
$\mathbf{C}_{inv}$ — inverse calculus for classical logic 187
$\mathbf{C}_{inv}$ for formulas in NNF 205
$\mathbf{C}_{tab}$ — tableau calculus for classical logic 190
$\mathbf{C}_{tab}$ for formulas in NNF 205
$\mathbf{C}^{\xi}_{inv}$ — free-variable inverse calculus for classical logic 198
calculus **249**
  complete **189**
  sound **189**
clause **222**
closed expression **185**
$\succ$-compact path sequent **242**
complementary literal **203**
complete calculus **189**
composition of substitutions **186**
conflict **239**
contraction **187**
contraction-free logics **255**

D

D **219**
D4 .... **219**
derivation **188**, **249**
diamond-separated **237**
disjunctive subformula **227**
dom(θ) — domain of substitution θ **185**
domain of substitution **185**

E

eigenvariable **187**
eigenvariable condition **187**
empty substitution **186**
expression **185**

F

fails (derivation in the set calculus) **249**
fair derivation **250**
finite rule property **184**
formula image **235**, **257**
free(E) — set of all free variables of E **185**
free(Γ) **197**
free immediate signed subformula **192**
free immediate subformula **203**
free signed subformula **192**
free subformula **203**

G

G — goal formula **234**
G-ordering **239**
$\mathbf{G}_{tab}$ — tableau calculus for Grishin's logic 255
goal **184**
goal formula **234**
goal object **249**
goal signed formula **184**
Grishin's logic 255
ground expression **185**
grounding substitution **185**

I

$\mathbf{I}_{inv}$ — inverse calculus for intuitionistic logic 211
$\mathbf{I}_{tab}$ — tableau calculus for intuitionistic logic 210
$\mathbf{I}^{\xi}_{inv}$ — free-variable inverse calculus for intuitionistic logic 214
immediate signed subformula **192**
immediate subformula **203**
inference **188**
inference rule **249**
  invertible **189**
inference system **249**
initial set of objects **249**
instance **196**
intuitionistic sequent **209**
invertible rule **189**
$\mathbb{P}^{G}_{inv}$ **245**
$\mathbb{P}^{G,\succ}_{inv}$ **246**
$\{\mathbb{P}^{G,\succ}_{inv}\}$ — $\mathbb{P}^{G,\succ}_{inv}$ with prefix-subsumption **251**

K

$\mathbf{K}_{inv}$ — inverse calculus for $\mathbf{K}$ **218**
$\mathbf{K}^{G}_{inv}$ — free-variable inverse calculus for $\mathbf{K}$ **219**
$\mathbf{K}_{tab}$ — tableau sequent calculus for $\mathbf{K}$ **217**
K4 .... **219**

L

limit **250**
linear representation of derivation **206**
literal **203**
  complementary **203**
literal path **234**

M

mgu(E_1, E_2) — most general unifier . **197**
mgu(σ, τ) — most general unifier of substitutions σ, τ **197**
modal length **237**
model **186**
more general substitution **196**
most general unifier **197**
  of substitutions **197**
multiset **185**
multiset difference **185**
multiset extension **248**

N

name **221**
name calculus for G **222**
negation normal form **203**
NNF **203**

O

objects **249**
G-ordering **239**
ordering procedure **240**

P

parametric signed formulas **188**
$\mathbb{P}^{G}_{tab}$ **235**
path
  for first-order logic **256**
  for propositional modal logic **234**
  in formula **234**
    first-order case **256**
path calculi **233**
path calculus **235**
path name **257**
path sequent **235**
prefix **237**
  proper **237**
prefix-free **237**
prefix-free path sequent **258**
prefix-subsumes **251**
prefix-subsumption rule **251**
principal signed formula **188**
proper prefix **237**

R

$ran(\theta)$ — range of substitution θ **185**
range of substitution **185**
rectified formula **195**
rectified sequent **195**
rectified signed formula **195**
redundancy criteria **232**
redundancy criterion 206
redundant inferences **232**
redundant sequents **232**
refutation **188**
relevant substitution **195**
renaming **197**

renaming away **197**
renaming rule **197**
representation via a free signed
subformula **195**
represents **195**
respect a G-ordering
for $\mathbb{P}^G_{inv}$ **246**
for $\mathbb{P}^G_{tab}$ **242**
restriction of substitution **196**
$Rules_G$ **224**

S

S4 .... **219**
satisfiable **186**
satisfiable sequent **189**
saturation algorithm 206
sequent **181**
of $\mathbf{C}^{\xi}_{inv}$ **196**
set calculus **249**
side signed formulas **188**
sign **181**
signed formula **181**
simultaneous most general unifier
of substitutions **213**
sound calculus **189**
subformula on path **234**
first-order case **256**
subformula property **182**, **194**, **234**
submultiset **185**
substitution **185**
subsumption **248**
for path sequents **258**
succeeds (derivation in the set calculus) **249**

T

T **219**
tableau path calculus for Grishin's logic **257**
term free for x in E **186**
trace **252**
true formula **186**
true sequent **186**

U

unifiable **196**
unifier **196**
Universal Recipe of Automated
Deduction **184**

V

$var(E)$ — set of all variables of E **185**
$var(G, \pi)$ **257**
variable range of substitution **185**
variable sequence on a path in G **257**
variable-disjoint **197**
variables of sequent in $\mathbf{C}^{\xi}_{inv}$ **197**
variant
of sequent in $\mathbf{C}^{\xi}_{inv}$ **197**
$vran(\theta)$ — variable range of θ **185**

W

weakly unifiable **197**