GENTZEN-TYPE SYSTEMS AND RESOLUTION RULE
PART II. PREDICATE LOGIC

G. MINTS

§0. Introduction. This paper is a sequel to Mints [17] where complete resolution-type calculi were constructed for several propositional modal logics including S5, S4, T, and K. Here we extend this to the predicate logic using the same method, which provides a general scheme for transforming a cutfree Gentzen-type system into a resolution-type system preserving the structure of derivations. This is a direct extension of the method introduced by Maslov [10] for the classical predicate logic. To make this paper self-contained, we recapitulate some material from Mints [17], i.e., Part I.

The main idea of Maslov's method can be summarized as follows. Resolution derivation of the goal clause g from a list X of input clauses can be obtained as the result of deleting X from the Gentzen-type cutfree derivation of the sequent $X \Rightarrow g$.

We show here how to treat predicate logics for which cutfree formulations with the subformula property are known. These include intuitionistic logic and quantified modal systems S4, T, K4, K, and only predicate logic S4 is presented in detail. Our resolution formulation of the predicate logic S5 illustrates here the treatment of systems possessing cutfree formulations in terms of semantic tableaux. We do not consider here the resolution formulation of the intuitionistic predicate logic since such formulation containing a device to avoid Skolemization was presented in Mints [13], [14].

Resolution method for a formal system C is determined by specifying

(i) a class of formulas called clauses,

(ii) a method of reduction of any formula F of the system C to a finite list X_F of clauses,

(iii) an inference rule (or rules) R called the resolution rule for deriving clauses, and

(iv) a derivation process by forward chaining so that all derivable objects are consequences of initial clauses and garbage removal from the search space is possible.

The resolution method is said to be sound and complete iff for any formula F the derivability of F in C is equivalent to derivability of the goal clause g from X_F using the rule R.

For systems based on classical logic the goal can be taken to be the empty clause $\emptyset$ (constant false). Indeed derivability of an atom g is equivalent to derivability of $\emptyset$ from the negation $\sim g$. There are several important features of the standard resolution method for classical logic which are highly desirable for any extension to the non-classical case deserving the name of resolution.

(i) Clauses should be much simpler than formulas in general with respect to complexity measure suitable for given systems C. In the classical case standard clauses are quantifier-free (with implicit universal quantification) disjunctions of literals. So there is neither nesting of Boolean connectives (in the propositional case) nor alternating quantifiers. For the intuitionistic propositional calculus clauses were defined in [13], [14] as implications with nesting at most 2. In the modal case one cannot deal only with clauses of modal depth 1 (except in S5), unless PSPACE = co-NP. Indeed, for most non-classical propositional logics below S5 the derivability problem is PSPACE-complete, but for formulas of depth 1 it is in co-NP. Nevertheless *non-initial* derivable clauses are disjunctions of modal literals, i.e., of the form l, $\Box l$, $\Diamond l$, where l is a variable or its negation.

(ii) Reduction of an arbitrary formula F to the form $X_F \rightarrow g$, where X_F is a finite list of clauses and g is a goal (propositional variable or $\emptyset$) should be much easier than the decision problem for the systems C in question. Some authors advise using distributivity for transforming classical formula to a set of clauses. Such a procedure is efficient for relatively small formulas and sometimes allows one to restrict the search space, especially for the classical predicate logic. On the other hand, this procedure is exponential in the worst case, that is, it has the same order of complexity as the existing decision procedures for classical propositional logic. Moreover, it destroys the structure of the original formula and is not applicable to non-classical systems, since they do not allow Skolemization.

We use instead a familiar depth-reducing transformation by introducing new propositional variables. It is linear in time, universally applicable, and preserves the structure of the original formula. The list of relations defining new variables can be considered as a new encoding of the original formula or as a presentation of the data structure of its subformulas.

(iii) It is natural to require that the resolution rule R for a given system be as close as possible to the standard resolution rule for the classical propositional calculus. For propositional systems based on classical logic we were able to preserve this rule completely. Differences between various modal systems were expressed by special rules for handling modalities, which can be used only together with the resolution rule, and so can be considered to be analogues of factorization or unification for the classical resolution.

For *predicate logic* we use ideas from Mints [13], [14] and Zamov [20], where resolution-type systems for predicate logics were formulated for non-Skolemized formulas. Here reduction of the formula-depth also plays an essential part.

(iv) Our requirement that the inference process should proceed by forward chaining corresponds to Maslov's [11] distinction between local methods (such as resolution or Maslov's inverse method) and global methods (such as semantic tableau methods with introduction of dummy variables and finding their values by searching through closure conditions for all branches of the semantic table). Both resolution and Maslov's inverse method work by deriving consequences from the negation of the original formula. This allows one to use subsumption:

if C is derived, then in most cases one can throw away C' when $C \to C'$ is valid, and so save space and time.

Reducing formulas to resolution (clause) form by introducing new predicates can lead to considerable growth of the search space due to resolution over literals containing these variables. One can restrict this disadvantage using the connection between resolution and Gentzen-type derivation discovered by Maslov's [10] and Maslov's [8] idea of decomposition (*razbivka*) of a formula. The latter simulates to a certain degree introduction of new variables. We propose a *strategy* which for the propositional case can be roughly formulated as follows: if a variable P was introduced to replace an occurrence of a given sign, then no clause containing occurrences of P with opposite sign can be derived by the resolution rule. Related ideas were employed by Voronkov. References to other work were given in Part I.

We begin with the resolution formulation due to N. Zamov of the first-order predicate logic which does not require Skolemization. The main new feature of Zamov's formulation is the presence of initial clauses $(\exists y)(L_1 \vee \cdots \vee L_n)$ and corresponding resolution rule

$$(R_\exists) \qquad \frac{(\exists y)(L \vee M);\ {\sim}L' \vee N}{(M \vee N)\sigma} \qquad \sigma = \mathrm{MGU}(L, L')$$

where σ does not contain substitution for y and the result $(M \vee N)\sigma$ does not contain y. It is this formulation which will be generalized to the modal case. The proofs will follow the pattern of Part I (i.e., Mints [17]) which used ideas by Maslov.

§**1. Classical predicate calculus.** Material presented in this section is partly familiar from the literature. The aim of this exposition is to collect it, streamline the proofs, and prepare for further treatment of more complicated systems. New material here is a Maslov-type completeness proof and treatment of introduced predicates in Section 1.5.

We begin in Section 1.1 with the extended language of clauses [Zamov 20] and show that it is sufficiently general in the sense of the introduction, i.e., that any formula can be reduced to a system of clauses in linear time and space. A cutfree Gentzen-type system GK for classical logic together with a normal form theorem for derivations is described in the Section 1.2. Section 1.3 gives a detailed description of the resolution strategy corresponding to Gentzen-type derivability and a structure-preserving translation GR from GK into the resolution system. Section 1.4 is devoted to inverse translation. Section 1.5 contains an example of treatment of strategies under this approach.

1.1. Non-Skolemized clauses. Recall that in the usual formulation of the resolution method derivable objects are *clauses* which are disjunctions $L_1 \vee \cdots \vee L_k$ $(k \geq 0)$ of *literals* i.e., of atoms (propositional letters) $p, q, r, p_1, \ldots$ and their negations ${\sim}p, {\sim}q, \ldots$. Here $t_1, \ldots, t_k$ are *terms* constructed from variables by function symbols. Clauses are treated modulo order and number of occurrences of literals. $\emptyset$ means the empty clause (interpreted as the constant *false*). Complement $\sim L$ of a literal L is defined in a standard way so that $\sim(\sim p) = p$.

We will preserve the same class of derivable objects, i.e., of possible results of derivations, but extend the total set of clauses.

Initial clauses are disjunctions of literals as well as the formulas of the form $(\exists y)(L_1 \vee \cdots \vee L_k)$ where $k \geq 2$ and $L_1, \ldots, L_k$ are literals. We shall be interested in derivability relations $X \vdash C$, where X is a set of initial clauses and C is a clause. Recall that free individual variables in initial clauses are understood to be universally quantified.

Recall the following well-known fact.

THEOREM 1.1. *For any propositional formula F a set X_F of initial clauses of length < 3 can be constructed linearly in F by introduction of new variables such that F is valid iff X_F is inconsistent, i.e., the sequent $\forall X_F \Rightarrow \emptyset$ is derivable.*

Proof. The main idea is to introduce new predicate variables for subformulas of F by equivalences; for example replace subformula $P(x) \vee Q(x)$ by $X(x)$, and express equivalence $\forall x(X(x) \leftrightarrow P(x) \vee Q(x))$ by the set of clauses $\forall x(\sim X(x) \vee P(x) \vee Q(x))$, $\forall x(\sim P(x) \vee X(x))$, $\forall x(\sim Q(x) \vee X(x))$. Then take as X_F the union of introduced sets of clauses completed by the negation of the predicate letter introduced for the formula F itself.

More precisely, assign to every non-atomic subformula A of F a new predicate variable P_A with the number of arguments equal to the number of free individual variables in A. Define A^* to be A for atomic A, and to be the result of replacing the immediate non-atomic subformulas B of A by $P_B(\boldsymbol{y})$, where $\boldsymbol{y}$ is the list of free individual variables in B. Put

$$E_A \equiv (P_A(\boldsymbol{y}) \leftrightarrow A^*) \tag{1}$$

where $\boldsymbol{y}$ is the list of all individual variables free in A.

We express E_A as set of (universal closures) of clauses $C_A = C_A^+ \cup C_A^-$, where C_A^+ corresponds to the implication $A^* \to P_A(\boldsymbol{y})$, and C_A^- corresponds to inverse implication $P_A(\boldsymbol{y}) \to A^*$. Assuming to simplify notation that immediate subformulas of A are non-atomic and free individual variables of immediate subformulas are the same, we make the following definitions:

If $A \equiv B \,\&\, D$ then

$$\begin{aligned} C_A^+ &\equiv \{\sim P_B(\boldsymbol{y}) \vee \sim P_D(\boldsymbol{y}) \vee P_A(\boldsymbol{y})\} \\ C_A^- &\equiv \{\sim P_A(\boldsymbol{y}) \vee P_B(\boldsymbol{y}), \sim P_A(\boldsymbol{y}) \vee P_D(\boldsymbol{y})\}. \end{aligned} \tag{$2_\&$}$$

If $A \equiv B \vee D$ then

$$\begin{aligned} C_A^+ &\equiv \{\sim P_B(\boldsymbol{y}) \vee P_A(\boldsymbol{y}), \sim P_D(\boldsymbol{y}) \vee P_A(\boldsymbol{y})\} \\ C_A^- &\equiv \{\sim P_A(\boldsymbol{y}) \vee P_B(\boldsymbol{y}) \vee P_D(\boldsymbol{y})\}. \end{aligned} \tag{$2_\vee$}$$

If $A \equiv \sim B$ then

$$\begin{aligned} C_A^+ &\equiv \{P_A(\boldsymbol{y}) \vee P_B(\boldsymbol{y})\} \\ C_A^- &\equiv \{\sim P_A(\boldsymbol{y}) \vee \sim P_B(\boldsymbol{y})\}. \end{aligned} \tag{$2_\sim$}$$

If $A \equiv \forall xB$ then

$$C_A^+ \equiv \{(\exists x)(\sim P_B(x,\boldsymbol{y}) \vee P_A(\boldsymbol{y})\}$$
$$C_A^- \equiv \{\sim P_A(\boldsymbol{y}) \vee P_B(x,\boldsymbol{y})\}. \qquad (2_\forall)$$

If $A \equiv (\exists x)B$ then

$$C_A^+ \equiv \{P_A(\boldsymbol{y}) \vee \sim P_B(x,\boldsymbol{y})\}$$
$$C_A^- \equiv \{(\exists x)(\sim P_A(\boldsymbol{y}) \vee P_B(x,\boldsymbol{y}))\}. \qquad (2_\exists)$$

If one of the subformulas, say B, is atomic we write B instead of $P_B(\boldsymbol{y})$, etc. Note that unusual clauses containing existential quantifiers arise in positive clauses for $\forall$-quantifiers and in negative clauses for $\exists$-quantifiers, i.e., exactly in the situations where Skolemization is used in the standard approach to resolution.

Now put:

$$Y_F = \{C_A : A \text{ is non-atomic subformula of } F\} \qquad (3)$$

$$X_F = Y_F \cup \{\sim P_F(\boldsymbol{c})\} \qquad (4)$$

where $\boldsymbol{c}$ is a list of new constants and P_F is the predicate assigned to F. Inconsistency of X_F is equivalent to the derivability of

$$\forall Y_F \vdash P_F(\boldsymbol{c}) \qquad (5)$$

1. Derivability of (5) implies derivability of F. Indeed, substitution of A for P_A in (5) gives

$$Y_F{}' \rightarrow F(\boldsymbol{c}) \qquad (6)$$

where all formulas in Y' are easily derivable, since they result (by decomposition into clauses) from the formulas of the form $B \rightarrow B$. So (6) implies $F(\boldsymbol{c})$ and substituting back free variables $\boldsymbol{x}$ of F for $\boldsymbol{c}$ we have $F(\boldsymbol{x})$, i.e., F.

2. Derivability of F implies derivability of (5). Assuming for simplicity that F is closed, we derive

$$F \rightarrow (\forall Y_F \rightarrow P_F) \qquad (7)$$

from

$$\forall Y_F \rightarrow (P_F \leftrightarrow F) \qquad (8)$$

which is obtained by repeated use of the replacement of equivalents

$$\forall \boldsymbol{x}(A \leftrightarrow B) \rightarrow (G[A] \leftrightarrow G[B]) \qquad (9)$$

where $\boldsymbol{x}$ contains all free variables of A, B, and G is any formula. This concludes the proof of the theorem.

Example 1. Let $F \equiv (\exists x)(\forall y)(P(x) \vee \sim P(y))$. Then denoting $P(x) \vee \sim P(y)$ by 1 and $\forall y(P(x) \vee \sim P(y))$ by 2 we have

$$E_1 \equiv P_1(x,y) \leftrightarrow (P(x) \vee \sim P(y)),$$
$$E_2 \equiv P_2(x) \leftrightarrow \forall y P_1(x,y),$$
$$E_F \equiv P_F \leftrightarrow (\exists x)P_2(x)$$

and so

$$X_F \equiv \{ \sim P_1(x,y) \vee P(x) \vee \sim P(y), \sim P(x) \vee P_1(x,y), P(y) \vee P_1(x,y), \sim P_2(x) \vee P_1(x,y), (\exists y)(\sim P_1(x,y) \vee P_2(x)), (\exists x)(\sim P_F \vee P_2(x)), \sim P_2(x) \vee P_F, \sim P_F\}.$$

1.2. Gentzen-type predicate calculus GK. Consider a Gentzen-type formulation of the classical predicate calculus suitable for pruning superfluous formulas (cf. below). Its derivable objects are *sequents* $X \Rightarrow Y$, where X, Y are finite (possibly empty) lists of formulas of the language considered. The order of formulas in X, Y will always be disregarded.

Gentzen-type system GK. Axioms: $A \Rightarrow A$.

Inference rules:

$$\frac{A_1, X_1 \Rightarrow Y_1; \ldots; A_n, X_n \Rightarrow Y_n}{A_1 \vee \cdots \vee A_n, X \Rightarrow Y} \quad (\vee \Rightarrow)$$

where $X_1 \cup \cdots \cup X_n = X$, $Y_1 \cup \cdots \cup Y_n = Y$.

$$\frac{X \Rightarrow Y, A_1, \ldots, A_m}{X \Rightarrow Y, A_1 \vee \cdots \vee A_n} \; (m \leq n)(\Rightarrow \vee) \qquad \frac{X \Rightarrow Y}{X', X \Rightarrow Y, Y'} \; \text{(thinning)}$$

$$(\Rightarrow \sim)\frac{A, X \Rightarrow Y}{X \Rightarrow Y, \sim A} \qquad \frac{X \Rightarrow Y, A}{\sim A, X \Rightarrow Y}(\sim\Rightarrow)$$

$$(\forall \Rightarrow)\; \frac{A[x := t], (\forall x A)^0, X \Rightarrow Y}{\forall x A, X \Rightarrow Y} \qquad \frac{X \Rightarrow Y, A[x := b]}{X \Rightarrow Y, \forall x A} \; (\Rightarrow \forall)$$

$$(\exists \Rightarrow)\; \frac{A[x := b], X \Rightarrow Y}{(\exists x)A, X \Rightarrow Y} \qquad \frac{X \Rightarrow Y, ((\exists x)A)^0, A[x := t]}{X \Rightarrow Y, \; (\exists x)A} \; (\Rightarrow \exists)$$

where superscript 0 means possible absence of the formula. So in fact we have two versions of the rule $(\forall \Rightarrow)$:

$$\frac{A[x := t], X \Rightarrow Y}{\forall x A, X \Rightarrow Y} \qquad \frac{A[x := t], \forall x A, X \Rightarrow Y}{\forall x A, X \Rightarrow Y}$$

and similarly for $(\Rightarrow \exists)$.

Let us fix terminology concerning Gentzen-type systems. In the sequent $X \Rightarrow Y$ the *left-hand side* X and *right-hand side* Y are sometimes called *antecedent* and *succedent*. In each inference rule the sequent written under the line is the conclusion, and the sequents over the line are premises. The formula shown explicitly in the conclusion, for example $A_1 \vee \cdots \vee A_n$ in $(\vee \Rightarrow)$ or X', Y' in (thinning), is the *main formula*, the formulas shown explicitly in the premises, for example $A_1, \ldots, A_n$ in the rule $(\vee \Rightarrow)$, are *side* formulas, and the remaining formulas, for example $X_1, Y_1, \ldots, X_n, Y_n, X, Y$ in $(\vee \Rightarrow)$ or X, Y in (thinning), are *parametric* formulas.

Recall the following facts from predicate logic.

THEOREM 1.2. *(a) Predicate formula F is valid iff the sequent $\Rightarrow F$ is derivable in GK.*

(b) The derivation of a sequent S uses only rules for connectives occurring in S, or more precisely, the succedent and antecedent rules corresponding to positive or negative occurrences of connectives.

(c) A list X of formulas is inconsistent iff the sequent $X \Rightarrow$ (with empty right-hand side) is derivable in GK.

(d) All thinning inferences can be moved downward (with possible deletion of some formulas and sequents) so that the thinning rule occurs only immediately preceding the last sequent of the derivation.

(e) Propositional inferences can be moved downward so that such inference L occurs only in a series of propositional inferences immediately preceding the last sequent of the derivation or immediately above the inference L' having the main formula of L as its side formula.

Proof. Cf. Kleene [4], [5].

A derivation satisfying the conditions in (d) (respectively in (e)) of Theorem 1.2 is called *pruned* (or p-*inverted*, respectively), and the operation of moving inferences downward mentioned there is called *pruning* (or p-*inversion*, respectively).

1.3. Resolution calculus corresponding to GK. Operation GR. Let us return to the clausal formulation (cf. Theorem 1.1). We consider a deduction relation $X \vdash C$ where X is a set of initial clauses and C is a clause. A positive (negative) occurrence of a predicate P in $X \vdash C$ is an occurrence in $P(t_1, \dots, t_n)$ (in $\sim P(t_1, \dots, t_n)$, respectively) as a member of C and in $\sim P(t_1, \dots, t_n)$ (in $P(t_1, \dots, t_n)$, respectively) as a member of one of the clauses in X.

Substitution is an expression of the form $[x_1 := t_1, \dots, x_n := t_n]$ where t_i are terms, x_i are distinct variables which do not occur in t_i. If this substitution is denoted by σ, then the result $E\sigma$ of its execution is obtained by replacing all occurrences of x_i in E by t_i, $i = 1, \dots, n$.

The *unifier* of the expressions E, F is a substitution σ unifying E and F, i.e., such that $E\sigma = F\sigma$. The *most general unifier* MGU(E, F) is the simplest unifier, that is $E\sigma' = F\sigma'$ implies $\sigma' = \sigma''\mathrm{MGU}(E, F)$ for some substitution σ''. The substitution in the right-hand side of the last equation is the result of the successive execution of MGU(E, F) and σ''. MGU($E_1, \dots, E_n$) is the most general unifier of all expressions $E_1, \dots, E_n$.

The general formulation of the resolution rule is

$$(R) \qquad \frac{L \vee E;\ \sim L' \vee D}{(E \vee D)\sigma} \qquad \sigma = \mathrm{MGU}(L, L')$$

where σ is the most general unifier of the literals L, L'. Alphabetic renaming of free variables is assumed throughout. More precisely, inference according to the rule (R) and all resolution-like rules below, like $(R_\exists)$, (RP), $(RP_\exists)$—but not in propositional-type rules $(R'_\exists)$, $(RP'_\exists)$—includes implicitly such a renaming to make all variables in $L \vee E$ distinct from all variables in $\sim L' \vee D$.

This form of the resolution is complete together with the factorization rule (cf. below).

To deal with (initial) clauses beginning with ($\exists$) we add the following rule introduced by Zamov [20]:

$$(R_\exists) \qquad \frac{(\exists y)(L \vee E);\ {\sim}L' \vee D}{(E \vee D)\sigma} \qquad \sigma = \mathrm{MGU}_\exists(L, L')$$

where subscript $\exists$ in MGU means that an additional proviso is imposed: σ is the most general substitution unifying L, L' which does not contain element $y := t$ and such that the resolvent $(E \vee D)\sigma$ does not contain y. It is understood as always that σ does not introduce collision of variables (here with the quantifier $(\exists y)$), i.e., y does not occur in elements $x := t$ with x free in $L \vee E$.

In other words σ does not change variable y and can introduce y only into L'. This restriction corresponds to the proviso in $\exists$-elimination rule for natural deduction

$$\frac{E \to (\exists y)L;\ L \to D}{E \to D}$$

where y should not occur in E, D. This remark is made precise in the following statement, where $\forall$ means universal closure.

LEMMA 1.3. *Under proviso of the rule* $(R_\exists)$ *the formula*

$$\forall(\exists y)(L \vee E) \,\&\, \forall({\sim}L' \vee D) \to (E \vee D)\sigma$$

is derivable, so the rule is sound.

Proof. We have by $\forall$-elimination: $\forall({\sim}L' \vee D) \to \forall y(({\sim}L' \vee D)\sigma)$. Since σ restricted to $L \vee E$ does not contain y we have $((\exists y)(L \vee E))\sigma \equiv (\exists y)((L \vee E)\sigma)$. By general properties of substitution

$$(L \vee E)\sigma \equiv L\sigma \vee E\sigma \quad \text{and} \quad ({\sim}L' \vee D)\sigma \equiv {\sim}L'\sigma \vee D\sigma.$$

Since $E\sigma$, $D\sigma$ do not contain y, equivalences

$$(\exists y)(L\sigma \vee E\sigma) \leftrightarrow (\exists y)L\sigma \vee E\sigma;\ \forall y({\sim}L'\sigma \vee D\sigma) \leftrightarrow \forall y\, {\sim}L'\sigma \vee D\sigma$$

are derivable. Finally, since σ is a unifier of L and L', we have $\forall y \,{\sim} L'\sigma \leftrightarrow {\sim}(\exists y)L\sigma$ and so

$$\forall(\exists y)(L \vee E) \,\&\, \forall({\sim}L' \vee D) \to ((\exists y)L\sigma \vee E\sigma) \,\&\, ({\sim}(\exists y)L\sigma \vee D\sigma) \to E\sigma \vee D\sigma$$

which is to be proved.

We introduce now a calculus for deriving $X \vdash C$ which has two kinds of axioms and three inference rules.

Resolution calculus RK. Axioms: Clauses belonging to the list X (input clauses) and $L \vee {\sim}L$ where the predicate symbol of the literal L occurs both positively and negatively in $X \vdash C$ (purity restriction).

Inference rules:

$$(RP) \qquad \frac{L_1 \vee \cdots \vee L_n \text{ (input)};\ {\sim}L'_1 \vee D_1; \ldots ; {\sim}L'_n \vee D_n}{(D_1 \vee \cdots \vee D_n)\sigma}$$

where $\sigma = \mathrm{MGU}(L_1, L'_1; \ldots; L_n, L'_n)$ i.e., σ is the most general substitution unifying each of the pairs (L_i, L'_i), $i = 1, \ldots, n$.

$$(RP_\exists) \qquad \frac{(\exists y)(L_1 \vee \cdots \vee L_n) \text{ (input)}; \ {\sim}L'_1 \vee D_1; \ldots; {\sim}L'_n \vee D_n}{(D_1 \vee \cdots \vee D_n)\sigma}$$

$\sigma = \mathrm{MGU}_\exists(L_1, L'_1; \ldots; L_n, L'_n)$, $n \geq 2$, where $\mathrm{MGU}_\exists$ means the proviso similar to the one in $R_\exists$ above: σ does not change the variable y and can introduce it only into $L'_1, \ldots, L'_n$.

$$(F) \qquad \frac{L_1 \vee \cdots \vee L_n \vee D}{L \vee D\sigma}$$

where $n > 1, \sigma = \mathrm{MGU}(L_1, \ldots, L_n)$ and $L = L_1\sigma = \cdots = L_n\sigma$.

Comments. The rule RP can be thought of as a series of n inferences according to the standard resolution rule (R) presented at the beginning of this section. It is closely connected with the clash rule [Chang and Lee, 1]. Our axioms of the form $L \vee {\sim}L$ are introduced to ensure a complete clash form of RP. It is easy to see that deletion of all such tautologies from a derivation in RK results in a derivation of the same clause as before by a series of inferences which can be thought of as multiple applications of the standard resolution rule to an initial clause. Assuming to simplify notation that the axiom premises of the form $L \vee {\sim}L$ are the last ones, we can write such a series of the standard resolution rules in the form:

$$(R_{si}) \qquad \frac{L_1 \vee \cdots \vee L_n \text{ (input)}; \ {\sim}L'_1 \vee D_1; \ldots; {\sim}L'_k \vee D_k}{(D_1 \vee \cdots \vee D_k \vee L_{k+1} \vee \cdots \vee L_n)\sigma}$$

$\sigma = \mathrm{MGU}(L_1, L'_1; \ldots; L_k, L'_k)$.

We can call this rule *semi-input* resolution. Recall that the input strategy of the standard resolution rule (R) is the requirement that at least one of the premises should be an input clause. This strategy is known to be incomplete [Chang and Lee, 1]. We shall see that the semi-input strategy is complete and corresponds to Gentzen-type derivability up to the structure of derivations for Skolemized formulas. For the case when the existential quantifier is present in clauses $(\exists y)D$ one should add a semi-input version of the rule $(RP_\exists)$.

Allowing axioms $L \vee {\sim}L$ without purity restriction would result in the admission of the substitution rule, since for every substitution σ we would have the following (RP)-inference:

$$\frac{L_1 \vee \cdots \vee L_n; \ L_1\sigma \vee {\sim}L_1\sigma; \ldots; L_n\sigma \vee {\sim}L_n\sigma}{(L_1 \vee \cdots \vee L_n)\sigma}$$

Purity restriction means that the substitution is allowed in all non-pure literals, i.e., ones which have a chance to be main literals in an axiom. An even more reasonable restriction is one used in Maslov's inverse method [Maslov, 8]: an axiom is a tautology of the form $L\sigma \vee {\sim}L'\sigma$ where L, ${\sim}L'$ are literals occurring positively in $X \vdash C$ and σ is their most general unifier with the natural $\exists$-proviso. Our completeness proof will in fact establish completeness of this restriction, but later we shall prove that all axioms of this kind are redundant (cf. Theorem 1.5.2).

One can get rid of the factorization rule by means of a familiar trick: replace resolution by its combination with factoring into the resolved literal. The completeness of this rule is proved by moving factorization downward in the derivation by resolution and factorization.

Proviso $n \geq 2$ in the rule $(RP_\exists)$ is made to simplify formulations, since it is satisfied for our initial clauses. We shall see that the completeness theorem is valid without this restriction and its proof requires only minor modification.

The calculus RK is used to derive clauses from some set X of input clauses. The leftmost premise of each of the rules (RP), $(RP_\exists)$, $(F_\exists)$ should be one of these input clauses and is called the *nucleus*, while remaining premises $\sim L'_i \vee D_i$ are *electrons*.

Notation $X \vdash C$ means that clause C is derivable in RK from the set X of clauses.

THEOREM 1.4. *A set X of clauses is inconsistent iff $\forall X \vdash \emptyset$ where $\forall$ means universal closure.*

This statement can be easily obtained from well-known results but we are interested here in presenting ideas of Maslov's [10] proof and the construction of Zamov [20] which we generalize in subsequent sections. Theorem 1.4 is obtained as a consequence of Theorem 1.2 (c), (d) and the properties of transformation GR of a Gentzen-type derivation d of a sequent $\forall X, Y \Rightarrow Y'$ (abbreviated $d : X, Y \Rightarrow Y'$) into a resolution derivation GR(d): $X \vdash \sim Y \vee Y'$, i.e., a derivation of $\sim Y \vee Y'$ from the initial clauses X. Here Y, Y' are lists of literals and $\sim Y \vee Y'$ is a clause consisting of members of Y' and complemented members of Y with obvious modifications when Y, Y' or both are empty. More precisely, GR(d) will derive a subclause of $\sim Y \vee Y'$, i.e., the result of deleting some (possibly no) literals from that clause.

We describe GR in detail to make possible references below.

In fact, GR(d) will be constructed in two steps. First we construct a derivation GR$'(d)$ from the substitution instances of initial clauses by the following two rules:

$$(RP')\frac{(L_1 \vee \cdots \vee L_n)\ \text{(input)};\ \sim L'_1 \vee D_1; \ldots ; \sim L'_n \vee D_n}{D_1 \vee \cdots \vee D_n}$$

where $\quad L_i\sigma = L'_i$ for $i = 1, \ldots, n$

$$(RP'_\exists)\frac{(\exists y)(L_1 \vee \cdots \vee L_n)\ \text{(input)};\ \sim L'_1 \vee D_1; \ldots ; \sim L'_n \vee D_n}{D_1 \vee \cdots \vee D_n}$$

where $L_i\sigma = L'_i$ for $i = 1, \ldots, n$ and the $\exists$-proviso should be satisfied: neither substitution σ nor the resolvent $D_1 \vee \cdots \vee D_n$ can contain the variable y. This corresponds to the propositional part of the familiar completeness proof for the resolution rule via Herbrand's theorem, but now even this step contains the quantifier rule $(RP'_\exists)$. The final derivation GR(d) will be produced by a standard lifting construction, that is, by replacing arbitrary substitutions in the rules (RP'), $(RP'_\exists)$ by the most general unifiers and adding factorizations when necessary.

Definition of the transformation GR′. Let $d : X, Y \Rightarrow Y'$ be a derivation in GK, X be a list of formulas of the form $\forall x_1 \cdots \forall X_n C$ where $n \geq 0$, C is an initial clause, and Y, Y' are lists of atoms. Then every sequent in d is of the form

$$X_1, Z \Rightarrow Z' \tag{10}$$

where X_1 is a list of instances of formulas from X and Z, Z' are lists of atoms. Replace each of these sequents by

$$\sim Z \vee Z' \tag{11}$$

and delete repetition of adjacent clauses, i.e., passages $C \,/\, C$. To transform the obtained figure into a derivation by rules (RP'), $(RP'_\exists)$ note that every inference in d belongs to one of the following types.

1. $(\vee \Rightarrow)$ with the main formula originating from a formula $\forall C$ in X, i.e., being an instance $C\theta$ of it where clause C does not contain the existential quantifier.
2. $(\vee \Rightarrow)$ with the main formula $C\theta[y := b]$ originating from a formula $\forall \boldsymbol{x}(\exists y)C$ in X, where θ is a substitution for variables in $\boldsymbol{x}$. Assume to simplify notation that in this case $b \equiv y$.
3. $(\sim\Rightarrow)$ with the main formula $C\sigma$ for $\forall C$ in X.
4. $(\sim\Rightarrow)$ with the main formula in Y.
5. $(\Rightarrow\sim)$ with the main formula in Y'.

Now for any $(\vee \Rightarrow)$-inference in d add the main formula of this inference as an additional premise (nucleus), i.e.,

$$\frac{L_1, X_1, Z_1 \Rightarrow Z'_1; \ldots; L_n, X_n, Z_n \Rightarrow Z'_n}{L_1 \vee \cdots \vee L_n, Z \Rightarrow Z'} \tag{12}$$

is transformed into

$$\frac{(\exists y)^0 C;\ \sim L_1 \vee \sim Z_1 \vee Z'_1; \ldots; \sim L_n \vee \sim Z_n \vee Z'_n}{\sim Z \vee Z'} \tag{13}$$

where $C\sigma \equiv L_1 \vee \cdots \vee L_n$ originates from $\forall C$ or $\forall(\exists y)C$; $(\exists y)^0$ stands for $\exists y$ in the second case and is empty in the first case.

For any $(\sim\Rightarrow)$-inference with conclusion (10) and main formula from X_1, add its main formula as an additional premise (nucleus), i.e., the inference

$$\frac{X, Z \rightarrow Z', A}{\sim A, X, Z \rightarrow Z'} \tag{14}$$

is transformed to

$$\frac{C\sigma;\ \sim Z \vee Z' \vee A :}{\sim Z \vee Z'} \tag{15}$$

where $\sim A \equiv C\sigma$.

This concludes the description of GR′(d).

Note. The conclusion of GR′ uses the restriction $n \geq 2$ for initial clauses $(\exists y)(L_1 \vee \cdots \vee L_n)$. It is easy to remove this restriction by treating the rule $(\exists \Rightarrow)$ with the main formula $(\exists y)L$

$$\begin{array}{rc} d' : & L, X, Y \Rightarrow Y' \\ \hline d : & (\exists y)L, X, Y \Rightarrow Y' \end{array}$$

in a special way. After constructing GR$'(d')$: $L, X \vdash \sim Y \vee Y'$ remove the initial clause L and add $\sim L$ to all clauses that depend on the removed L. This will give a derivation $X \vdash L \vee \sim Y \vee Y'$ and GR$'(d)$ is concluded by $(RP'_{\exists})$ with nucleus $(\exists y)L$. So most of the following results are preserved with their proofs without the restriction $n \geq 2$ for initial $(\exists)$-clauses.

THEOREM 1.5. *(a) If X is a set of formulas of the form $\forall x_1 \cdots x_n C$, where $n \geq 0$ and C are initial clauses, Y, Y' are lists of literals and $d : X, Y \Rightarrow Y'$ is a pruned p-inverted GK-derivation containing no thinnings then*

$$GR'(d) : X^{\forall} \vdash \sim Y \vee Y' \qquad \text{by the rules } (RP'), (RP'_{\exists}) \tag{16}$$

where $X^{\forall}$ is the result of dropping initial $\forall$-quantifiers from the formulas in X, or

$$X \text{ is empty and } Y = Y' \text{ is one and the same literal.} \tag{17}$$

(b) If X, C are lists of initial clauses, Y is a list of literals, and $\forall X, Y \Rightarrow C$ is GK-derivable, then $X \vdash (\sim Y \vee C)^-$ by the rules (RP'), $(RP'_{\exists})$ where minus means deletion of some (possibly no) literals from $\sim Y$, C.

(c) In particular GK-derivability of $\forall X \Rightarrow D$ for a clause D implies $X \vdash D^-$ and if X is inconsistent (i.e., $X \Rightarrow$ is GK-derivable), the $X \vdash \emptyset$.

Proof. (a) Apply induction on (the number of inferences in) the pruned derivation d of sequent X, $Y \Rightarrow Y'$.

Induction base. If Y' is X then Y is empty and $\sim Y \vee Y' = Y' = X$ is the initial clause. If Y' is Y the X is empty and we have (17).

Induction step. Only the rules $\sim\Rightarrow$, $\Rightarrow\sim$, $\vee \Rightarrow$, $\forall \Rightarrow$ and final thinning can be applied in the derivation.

Case 1. The inference $\Rightarrow\sim$: $A, X, Y \Rightarrow Y'/X, Y \Rightarrow Y', \sim A$ is transformed into repetition $\sim Y \vee Y' \vee \sim A / \sim Y \vee Y' \vee \sim A$ (recall that we identify clauses differing only by permutation).

Case 2. The rule $\sim\Rightarrow$. If the main formula does not belong to X, then $\sim\Rightarrow$ is transformed into repetition since $\sim\sim A = A$. In the opposite case the $\sim\Rightarrow$ is transformed into the figure (15) which is the application of RP. The only non-trivial case is where (7) holds for the premise of the rule. Then the electron $\sim Y \vee Y$ should satisfy the purity restriction. Indeed, our derivation is pruned, so any atom (including Y) is traceable to an axiom $Y \Rightarrow Y$ and the purity restriction is satisfied.

Case 3. The rule $\vee \Rightarrow$. It has the form (12) and is transformed into (13) where $(\exists y)$ is absent and C is $L_1 \vee \cdots \vee L_n$. The purity condition for the premises of (13) satisfying (17) is valid for the same reason as in case 2.

Case 4. The rule $(\exists \Rightarrow)$. This is the main new point of our proof compared to Theorem 1.3 of Part I. The main formula of the rule $\exists \Rightarrow$ is the initial clause $(\exists y)C$, so C should be a disjunction. Since the given Gentzen-type derivation d

is p-inverted, its final part is of the following form:

$$\begin{array}{ll} & \underline{L_1, Z_1, Y_1 \to Y_1'; \ \ldots \ ; L_n, Z_n, Y_n \Rightarrow Y_n'} \\ d': & \quad\underline{L_1 \lor \cdots \lor L_n, Z, Y \Rightarrow Y'} \\ d: & \ (\exists y)(L_1 \lor \cdots \lor L_n), Z, Y \Rightarrow Y' \end{array}$$

By the definition GR$'(d')$ ends in RP':

$$(RP') \qquad \frac{L_1 \lor \cdots \lor L_n; {\sim}L_1 \lor {\sim}Y_1 \lor Y_1'; \ldots ; {\sim}L_n \lor {\sim}Y_n \lor Y_n'}{{\sim}Y \lor Y'} \tag{18}$$

In view of the proviso for the rule $(\exists \Rightarrow)$ the variable y does not occur in Y, Y', so prefixing $(\exists y)$ to the nucleus $L_1 \lor \cdots \lor L_n$ in (18) transforms our (RP) into a correct application of $(RP_\exists')$.

Note for the following that the $(RP_\exists')$-inferences in GR$'(d)$ exactly correspond to $(\exists \Rightarrow)$-inferences in d.

Case 5. The rule $(\forall \Rightarrow)$.

$$\begin{array}{ll} d': & \underline{C[x := t], \forall x C, X, Y \Rightarrow Y'} \\ d: & \qquad \forall x C, X, Y \Rightarrow Y' \end{array}$$

The derivation GR$'(d')$ differs from GR$'(d)$ only by replacement of some nuclei $C[x := t]$ in rules (RP'), $(RP_\exists')$ by C with corresponding addition of the element $[x := t]$ to the substitution σ. The only thing to check is the preservation of the $\exists$-proviso in each $(RP_\exists')$-inference

$$(RP_\exists') \qquad \frac{(\exists y)(L_1 \lor \cdots \lor L_n); \ {\sim}L_1' \lor D_1; \ldots ; {\sim}L_n' \lor D_n}{D_1 \lor \cdots \lor D_n} \qquad L_i\sigma = L_i'$$

If its nucleus is different from $C[x := t]$, this inference is not changed at all. Otherwise the nucleus is changed to C, i.e., the substitution $x := t$ is removed from $L_1 \lor \cdots \lor L_n$ and added to σ. As was just noted, the variable y of $(RP_\exists')$ in GR$'(d')$ is an eigenvariable of some $(\exists \Rightarrow)$-inference in d'. So it cannot occur free in the last sequent of d', in particular in the substitution $x := t$, and new substitution in $(RP_\exists')$ does not contain y as required.

This concludes the proof of (a).

(b) Let X, C be lists of clauses, $C = C_1, \ \ldots \ , C_n$. Let Y be a list of literals, and $\forall X, Y \Rightarrow C$ be GK-derivable. Let $C^\frown$ be the result of replacing all disjunctions in C by commas. Then $\forall X, Y \Rightarrow C^\frown$ is GK-derivable (use cuts with derivable sequents $C_i \Rightarrow C_i^\frown$) and ${\sim}Y \lor C \equiv {\sim}Y \lor C^\frown$. The pruned p-inverted GK-derivation of $X, Y \Rightarrow C^\frown$, which exists by Theorem 1.2(d), contains thinning only at the very end: $\forall X^-, Y^- \Rightarrow C^{\frown -}$ / $\forall X, Y \Rightarrow C$. Applying part (a) of the present theorem to $\forall X^-, Y^- \Rightarrow C^\frown$, we have $X \vdash {\sim}Y^- \lor C^-$ as required in (b). Additional information is that taking subclauses corresponds to thinning inferences in Gentzen-type derivation.

Part (c) of the theorem immediately follows from (b).

Now we define GR(d) as the result of the lifting, i.e., moving substitutions maximally down the derivation GR$'(d)$. For any derivation

$$d : X \vdash C \text{ by the rules } (RP'), (RP_\exists')$$

we define by induction on d the derivation $d : X \vdash C^\frown$ in the system RP such that $C = C^\frown\theta$ for some substitution θ.

If C is in X then $d' = d$. If C is $L \vee \sim L$, then $C^\frown$ is $P(x_1, \dots, x_n) \vee \sim P(x_1, \dots, x_n)$ where P is the predicate symbol of L and $x_1, \dots, x_n$ are pairwise distinct variables.

Let d end in the rule $(RP'_\exists)$:

$$\frac{(\exists y)(L_1 \vee \cdots \vee L_n)(\text{input});\ \sim L'_1 \vee D_1 : \ldots ; \sim L'_n \vee D_n}{D_1 \vee \cdots \vee D_n} \qquad \begin{array}{l} L_i\sigma \equiv L'_i, \\ i = 1, \ldots, n \end{array}$$

By the induction assumption we have derivations $d'_i : X \vdash (\sim L'_i \vee D_i)^\frown$ for $i = 1, \dots, n$, such that $\sim L'_i \vee D_i \equiv (\sim L_i \equiv D_i)^\frown \theta_i$. Slightly abusing $^\frown{}^-$ notation we write $(\sim L'_i \vee D_i)^\frown \equiv \sim L'^\frown_i \vee D^\frown_i$. Note that $\sim L'^\frown_i$ can contain more than one literal, but it is unified in one literal $\sim L'_i$ by the substitution θ_i. So applying if necessary the factorization rule (F) we will assume that all $\sim L'^\frown_i$ $(i = 1, \dots, n)$ are literals. In view of the proviso for $(RP'_\exists)$ the substitution σ does not contain the variable y, and the substitutions θ_i do not introduce it into $D_1 \vee \cdots \vee D_n$. Renaming if necessary free variables in $\sim L'^\frown_i \vee D^\frown_i$, we can assume that renaming conditions are satisfied for $(\exists y)(L_1 \vee \cdots \vee L_n)$, $\sim L'^\frown_1 \vee D^\frown_1, \dots, \sim L'^\frown_n \vee D'^\frown_n$. Collecting θ_i into the common substitution θ we have

$$L_i\sigma\theta \equiv L_i\sigma \equiv L'_i \equiv L'^\frown_i\theta \equiv L'^\frown_i\ \sigma\theta.$$

So $\sigma\theta$ is a general unifier of all pairs $(L_i, L'_i), i = 1, \dots, n$ satisfying these conditions for the following $(RP_\exists)$-inference:

$$(RP_\exists) \qquad \frac{(\exists y)(L_1 \vee \cdots \vee L_n);\ \sim L'^\frown_1 \vee D^\frown_1; \ldots ; \sim L'^\frown_n \vee D^\frown_n}{(D^\frown_1 \vee \cdots \vee D^\frown_n)\sigma^\frown}$$

Since $\sigma^\frown$ is the most general unifier, we have $\sigma\theta \equiv \sigma^\frown\theta'$ for some substitution θ'.

Since $D^\frown_i\theta \equiv D^\frown_i\theta_i \equiv D_i$ $(i = 1, \dots, n)$, and $\sigma\theta$ on $D^\frown_1, \dots, D^\frown_n$ coincides with θ, we have $D_i = D^\frown_i\theta = D^\frown_i\sigma^\frown\theta'$ $(i = 1, \dots, n)$, so the former resolvent $D_1 \vee \cdots \vee D_n$ is indeed a substitution instance of the new resolvent. That concludes the treatment of the rule $(RP'_\exists)$.

If d ends in the rule (RP'), the treatment is similar, but easier since there are no $\exists$-restrictions. This concludes description of $\mathrm{GR}(d)$.

THEOREM 1.6. *(a) If X is a set of initial clauses, Y, Y' are lists of literals, and $d : \forall X, Y \Rightarrow Y'$ is a pruned p-inverted GK-derivation containing no thinnings, then GR(d) : $X \vdash \sim Y \vee Y'$ in the system RP, or X is empty and $Y = Y'$ is one and the same literal.*

(b) If X, C are lists of initial clauses, Y is a list of literals, and $\forall X, Y \Rightarrow C$ is GK-derivable, then $X \vdash (\sim Y \vee C)^-$ in the system RP where minus means subsumption.

(c) In particular GK-derivability of $\forall X \Rightarrow D$ for a clause D implies $X \vdash D^-$, and if X is inconsistent (i.e., $X \Rightarrow$ is GK-derivable), then $X \vdash \emptyset$.

The proof is the same as for Theorem 1.5 using operation GR instead of GR$'$.

1.4. Translation RG of semi-input resolution into Gentzen-type derivations. Operation RG will be applied to a derivation by semi-input resolution $d : X \vdash D$ of a clause D from a set of clauses X and arbitrary partition $D \equiv D_1 \vee D_2$ of D into (possibly empty) clauses D_1, D_2. The result $\mathrm{RG}(d, D_1, D_2)$ (which we shall usually write as $\mathrm{RG}(d)$) is a Gentzen-type derivation of the sequent $\forall X, \sim D_1 \Rightarrow D_2$ where $\sim D_1$ consists of complements of literals in D_1 (if any).

Definition of $\mathrm{RG}(d)$ by induction on d.

If $D_1 \vee D_2 = D$ is a member of X, then (assuming for simplicity that D_1, D_2 are both unit clauses), $\mathrm{RG}(d)$ is of the form

$$\dfrac{\dfrac{\dfrac{D_1 \Rightarrow D_1}{D_1, \sim D_1 \Rightarrow} \qquad D_2 \Rightarrow D_2}{D_1 \vee D_2, \sim D_1 \Rightarrow D_2}}{X, D_1 \vee D_2, \sim D_1 \Rightarrow D_2} \quad \text{(thinning)}$$

If $D_1 \vee D_2 = \sim L \vee L$ (in some order) then $\mathrm{RG}(d)$ is the obvious derivation from the axiom $L \Rightarrow L$.

If the last inference of d is RP, choose the partition in each of the premises in accordance with the partition of the conclusion, i.e., put literals from D_1 in the first part, and the literals from D_2 in the second part. Then the RP in question can be written as

$$\frac{L_1 \vee \cdots \vee L_n;\ \sim L'_1 \vee D_{11} \vee D_{12}; \ldots ; \sim L'_n \vee D_{n1} \vee D_{n2}}{(D_{11} \vee \cdots \vee D_{n1} \vee D_{12} \vee \cdots \vee D_{n2})\sigma \equiv (D_1 \vee D_2)\sigma}$$

Putting the resolved literals $\sim L'_1, \ldots, \sim L'_n$ in the first part of the partitioning and applying the inductive assumption we can construct GKp-derivations of the sequents

$$\forall X, L'_1, \sim D_{11} \Rightarrow D_{12}; \ldots ; \forall X, L'_n, \sim D_{n1} \Rightarrow D_{n2}.$$

After making substitution σ we can conclude $\mathrm{RG}(d)$ by the following $(\vee \Rightarrow)$ and $(\forall \Rightarrow)$-inferences taking into account that $L'_i\sigma \equiv L_i\sigma$:

$$\dfrac{\dfrac{\forall X, L_1\sigma, \sim D_{11}\sigma \Rightarrow D_{12}\sigma; \ldots ; \forall X, L_n\sigma, \sim D_{n1}\sigma \Rightarrow D_{n2}\sigma}{\forall X, (L_1 \vee \cdots \vee L_n)\sigma, \sim D_1\sigma \Rightarrow D_2\sigma} (\vee \Rightarrow)}{\forall X, \sim D_1\sigma \Rightarrow D_2\sigma} (\forall \Rightarrow) \qquad (19)$$

If the last inference of d is (F), one should make a factorizing substitution, erase superfluous copies of identical literals, and add (if necessary) the $\sim$ inferences to complete former axioms $L \Rightarrow L$ which became $\sim L, L \Rightarrow$ or $\Rightarrow \sim L, L$.

If the last inference of d is $(RP_\exists)$, the treatment is similar to the case of (RP). One has only to add $(\exists \Rightarrow)$-inference in the figure (19). The proviso of the rule $(\exists \Rightarrow)$ is satisfied in view of the proviso for the rule $(RP_\exists)$. This concludes the definition of $\mathrm{RG}(d)$.

THEOREM 1.7. *If $d : X \vdash D_1 \vee D_2$ then $RG(d, D_1, D_2) : \forall X, \sim D_1 \Rightarrow D_2$. In particular $d : X \vdash D$ implies $RG(d) : \forall X \Rightarrow D$.*

Proof by induction on d is in fact contained in the definition of RG(d).

Important note. Operations RG and GR (defined in the previous section) preserve much of the structure of the derivation. In particular there is a close correspondence between the $(\vee \Rightarrow)$, $(\exists \Rightarrow)$-inferences in a Gentzen-type derivation and (RP), $(RP_\exists)$-inferences in corresponding RP-derivation. This enabled Maslov [10] to transfer many results on strategies for one of these formulations to another one.

1.5. Completeness of strategies. We illustrate the use of the apparatus of the preceding section in proving completeness of two strategies. A much more general result was established by Maslov [9].

V. Neiman and V. Orevkov noted that hyperresolution is incomplete for input clauses with $\exists$, as the following example due to V. Orevkov shows:

$$(\exists x)P(x),\ \forall y(\sim P(y) \vee Q(y)), \forall u \sim Q(u) \vdash \emptyset$$

On the other hand, V. Orevkov noticed that it is possible to require that all $(RP_\exists)$-inferences follow (i.e., be situated below) all (RP)-inferences.

From now until the end of this section we will be interested in derivability relations $X \vdash g$ encoding (according to a refinement of Theorem 1.1) the derivability of a predicate formula F. It was noted very early that the introduction of new predicates for this encoding leads to an increase in the search space. We shall prove the completeness in the Skolemized case of a strategy combining hyperresolution with a device essentially restricting this defect.

Let us consider first a more economical encoding. We use the notation from the proof of Theorem 1.1, especially formulas (1), (2) and symbols C_A^+, C_A^-. Instead of including into the encoding set all clauses C_A^+, C_A^- as was done in (3) there, we include only C_A^+ if the replaced occurrence of A is positive, and C_A^- if the replaced occurrence is negative. Instead of the predicates P_A for non-atomic formulas A we use P_A^+, P_A^- respectively for positive and negative occurrences. Instead of the equivalence (1) we use implications

$$I_A^- \equiv (P_A^-(y) \to A); \qquad I_A^+ \equiv (A \to P_A^+(y)) \tag{21}$$

and put

$$Z_F = \bigcup \{I_A^\sigma; A \text{ is a non-atomic subformula having sign } \sigma \text{ in } F\}.$$

Note that it is not necessary to introduce new predicates for all non-atomic subformulas of F. For example one can treat literals as atoms, and encode multiple disjunction or conjunction of literals by a single predicate. The same is true for chains of negative quantifiers, for example for positive occurrences of $(\exists x)(\exists y) \sim(\forall z)$ or negative occurrences of $\forall x \forall y \sim(\exists z)\ldots$, etc. Similar optimization for positive quantifiers would be possible if we introduced a special resolution rule for chains of existence quantifiers.

We call the relation $Z_F \vdash P_F^+$ obtained in this way (possibly with optimization of the kind mentioned above) a *standard encoding* of the formula F.

Instead of the deductive equivalence stated in Theorem 1 we shall describe a more close connection between derivations of F and its standard encoding

$Z_F \vdash P_F^+$. Without loss of generality we consider closed formulas F since free variables can be replaced by new constants.

Definition. Let d be a GK-derivation of the formula F. Then d^c will be a derivation of the clause form of F

$$d^c : \forall Z_F \vdash P_F^+$$

constructed as follows. Replace each occurrence of non-atomic subformulas $A[\boldsymbol{x} := t]$ as a member of a sequent in d by $P_A^\sigma(t)$. Here $\boldsymbol{x}$ is the list of free variables A which are bound in F, and σ is $+$ if the replaced occurrence is in the succedent (i.e., is positive in the whole sequent); if the replaced occurrence of $A[\boldsymbol{x} := t]$ is in the antecedent, then put $\sigma \equiv -$. Then add $\forall Z_F$ to the antecedents of all sequents of the resulting figure except the uppermost ones, and make insertions to turn the figure into the derivation.

To simplify the description we assume that the formula A in any axiom $A \Rightarrow A$ of the derivation d is atomic. Then no insertion is made for thinnings. Suppose that L is a logic inference.

1. L is $\Rightarrow\sim$:

$$\frac{A[\boldsymbol{x} := t], X \rightarrow Y}{X \rightarrow Y, \sim A[\boldsymbol{x} := t]}$$

By the steps already described it is transformed in the figure:

$$\frac{P_A^-(t), X' \Rightarrow Y'}{X' \Rightarrow Y', P_{\sim A}^+(t)} \tag{22}$$

The presence of the positive occurrence of $\sim A$ means that the list Z_F of clauses describing new variables introduced for encoding of the formula F contains the clause $P_{\sim A}^+(\boldsymbol{x}) \vee P_A^-(\boldsymbol{x})$, and so the antecedent X' above contains its universal closure. Now we make an insertion transforming the figure (22) into the following deduction:

$$\cfrac{\cfrac{P_{\sim A}^+(t) \Rightarrow P_{\sim A}^+(t); \quad P_A^-(t), X' \Rightarrow Y'}{P_{\sim A}^+(t) \vee P_A^-(t), X' \Rightarrow Y', P_{\sim A}^+(t)}\ (\vee\Rightarrow)}{X' \rightarrow Y', P_{\sim A}^+(t)}\ (\forall\Rightarrow)$$

Note that the leftmost sequent is an axiom.

2. L is $\sim\Rightarrow$:

$$\frac{X \Rightarrow Y, A[t]}{\sim A[t], X \Rightarrow Y}$$

Using the corresponding clause in Z_F it is transformed into the following deduction:

$$\cfrac{\cfrac{\cfrac{P_{\sim A}^-(t) \Rightarrow P_{\sim A}^-(t)}{\sim P_{\sim A}^-(t), P_{\sim A}^-(t) \Rightarrow;}\ (\sim\Rightarrow) \quad \cfrac{X' \Rightarrow Y', P_A^+(t)}{\sim P_A^+(t), X' \Rightarrow Y'}\ (\sim\Rightarrow)}{\sim P_A^-(t) \vee \sim P_A^+(t), P_{\sim A}^-(t), X' \Rightarrow Y'}\ (\vee\Rightarrow)}{P_{\sim A}^-(t), X' \Rightarrow Y'}\ (\forall\Rightarrow)$$

3. Now we list the results of transforming $\vee$-rules, assuming to simplify notation that the $(\Rightarrow \vee)$ rule has the form $X \Rightarrow Y,\ A\ /\ X \Rightarrow Y,\ A \vee B$. It is

transformed into the deduction

$$\dfrac{\dfrac{X' \Rightarrow Y', P_A^+(t)}{\sim P_A^+(t), X' \Rightarrow Y'} \qquad P_{A\vee B}^+(t) \Rightarrow P_{A\vee B}^+(t)}{\dfrac{\sim P_A^+(t) \vee P_{A\vee B}^+(t), X' \Rightarrow Y', P_{A\vee B}^+(t)}{X' \Rightarrow Y', P_{A\vee B}^+(t)}}$$

The rule $(\vee \Rightarrow)$ is transformed into the deduction

$$\dfrac{\dfrac{P_{A\vee B}^-(t) \Rightarrow P_{A\vee B}^-(t)}{\sim P_{A\vee B}^-(t), P_{A\vee B}^-(t) \Rightarrow;} \qquad P_A^-(t), X_1' \Rightarrow Y_1'; \quad P_B^-(t), X_2' \Rightarrow Y_2'}{\dfrac{\sim P_{A\vee B}^-(t) \vee P_A^-(t) \vee P_B^-(t), P_{A\vee B}^-(t), X' \Rightarrow Y'}{P_{A\vee B}^-(t), X' \Rightarrow Y'}}$$

4. The rule $\Rightarrow \forall$:

$$\frac{X \Rightarrow Y, A[\boldsymbol{x} := \boldsymbol{t},\ y := b]}{X \Rightarrow Y, \forall y A[\boldsymbol{x} := \boldsymbol{t}]}$$

is transformed into the deduction:

$$\dfrac{\dfrac{X' \Rightarrow Y', P_A^+(\boldsymbol{t}, b)}{\sim P_A^+(\boldsymbol{t}, b), X' \Rightarrow Y';} \qquad P_{\forall y A}^+(\boldsymbol{t}) \to P_{\forall y A}^+(\boldsymbol{t})}{\dfrac{\sim P_A^+(\boldsymbol{t}, b) \vee P_{\forall y A}^+(\boldsymbol{t}), X' \Rightarrow Y', P_{\forall y A}^+(\boldsymbol{t})}{X' \Rightarrow Y', P_{\forall y A}^+(\boldsymbol{t})}}$$

using clause $\forall \boldsymbol{x}(\exists y)(\sim P_A^+(\boldsymbol{x}, y) \vee P_{\forall y A}^+(\boldsymbol{x}))$ and the rules $(\forall \Rightarrow)$, $(\exists \Rightarrow)$. The same proviso for the variable b is required in both cases.

5. The rule $(\forall \Rightarrow)$ is transformed similarly using clause $\forall \boldsymbol{x} \forall y(P_A^-(\boldsymbol{x}, y) \vee \sim P_{\forall y A}^-(\boldsymbol{x}))$.

6. The rules for $\exists$ are treated similarly using clauses $\forall \boldsymbol{x} \forall y(P_{(\exists y)A}^+(\boldsymbol{x}) \vee \sim P_A^+(\boldsymbol{x}, y))$ and $\forall \boldsymbol{x}(\exists y)(\sim P_{(\exists y)A}^-(\boldsymbol{x}) \vee P_A^-(\boldsymbol{x}, y))$.

This concludes description of the derivation d^c.

Let us describe a strategy for resolution derivation of the canonical encoding which corresponds to Gentzen-type derivation of the encoded formula F. Recall that each clause in the antecedent Z_F of the standard encoding of F belongs to a set I_A^σ for some non-atomic subformula A of F. So such a clause contains a unique literal beginning with the predicate P_A^σ for $\sigma \equiv +$ or $-$. We call it the *leading literal* of the clause.

The resolution inference

$$\frac{(\exists y)^0(L_1 \vee \cdots \vee L_n);\ \sim L_1' \vee D_1; \ldots ; \sim L_n' \vee D_n}{(D_1 \vee \cdots \vee D_n)\sigma}$$

with the nucleus $(\exists y)^0(L_1 \vee \cdots \vee L_n)$ from Z_F will be called *G-inference* (or *G-resolution*) if the electron $\sim L_i' \vee D_i$ corresponding to the leading literal of the nucleus is a tautology $\sim L_i' \vee L_i'$. In other words, the leading literal is in fact not resolved in G-resolution, but preserved in the conclusion in the form $L_i'\sigma$, i.e., possibly with some substitution.

THEOREM 1.8. *Let d be a GK-derivation of a predicate formula F. Then:*
(a) $d^c : Z_F \Rightarrow P_F^+$ is a GK-derivation of the canonical encoding of F;
(b) $GR(d^c) : Z_F \vdash P_F^+$ is the derivation of the canonical encoding of F by G-resolution.

Proof. Part (a) was verified in the definition of d^c.
Part (b) immediately follows from the following facts.
(b1) In each $(\vee \Rightarrow)$-inference in d^c with the main formula being the result of dropping quantifiers from a clause in Z_F, the premise containing the leading literal is an axiom for this literal. This is verified by inspection of the definition of d^c.
(b2) If a premise of an $(\vee \Rightarrow)$-inference in a GK-derivation d is an axiom for some side formulas, then the corresponding electron of the resolution rule in GR(d) is a tautology. This is verified by inspection of the definition of GR(d). This concludes the proof.

Let us now prove that G-resolution is compatible with the hyperresolution for the canonical encodings of Skolemized formulas F in *positive normal form*, i.e., constructed from literals by $\vee, \&, \exists$. Without this latter restriction even the encoding of $\sim\sim a \vee \sim a$:

$$(\sim a \vee \sim \underline{n}), n \vee \underline{d}, a \vee \underline{d} \vdash d$$

with leading literals underlined, does not have a hyperresolution G-derivation. The restriction to positive normal form is inessential for Skolemized formulas since elimination of implication and moving negation inside is done in linear time and preserves the structure of a formula.

THEOREM 1.9. *Hyperresolution together with G-resolution is complete for canonical encodings of formulas in positive normal form.*

The proof uses the idea employed in Section 1.5 of Part I. Note that for a formula F in positive normal form all leading literals from the clauses in Z_F are positive, since they correspond to non-atomic subformulas of F, and the latter occur in F positively. We call a clause $P \vee N$, where P (the positive part of the clause) consists of positive literals and N (negative part) of negative literals, to be *essentially negative* in a sequent $S \equiv (P \vee N, X \Rightarrow)$, if X contains as separate clauses the negations of all literals in P. For example, in the sequent $a \vee b \vee c, \sim a \vee b \vee c, \sim b, \sim c, b \vee \sim d \vee \sim e \Rightarrow$ the second clause and the last clause are essentially negative, but the first clause is not. Our strategy (call it essentially negative) allows us to apply $\vee \Rightarrow$ only with an essentially negative main formula, i.e., to analyze in the process of the proof search only essentially negative clauses.

LEMMA 1.10. *Essentially negative strategy is complete for propositional calculus.*

Proof (reproduced from Part I). It is sufficient to prove that each provable sequent $X \Rightarrow$ where X is a list of clauses, either contains a complementary pair p, $\sim p$, or contains an essentially negative clause of length > 1. Indeed, in the latter case we can apply ($\vee\Rightarrow$) bottom-up according to our strategy and diminish the length of the sequent. Assume for contradiction that X does not contain such a clause. Then each clause C in X of length > 1 contains a positive literal p_c such that $\sim p_c$ is not a member of X. Then the valuation making all p_c true validates all clauses of length > 1 and does not falsify any clause of length 1. Putting all the latter true validates X, which contradicts derivability of $X \Rightarrow$.

Proof of Theorem 1.9. Let X be the contradictory set of clauses and d be its derivation in GK according to essentially negative strategy. We show that GR(d) is an essentially negative derivation of $\emptyset$ from X according to G-strategy. Recall that any application of the rule (RP') in GR(d) results from an ($\vee\Rightarrow$)-inference in the derivation d. To simplify notation, assume that all ($\vee\Rightarrow$)-inferences in d are below all ($\sim\Rightarrow$)-inferences. This is easy to achieve by simply moving ($\sim\Rightarrow$)-inferences up to axioms. So all sequents in d except axioms have empty succedents. To fix notation suppose that $L_1, \dots, L_k$ in (RP') are negative, and $L_{k+1}, \dots, L_n$ are positive. Since d satisfies essentially negative strategy, all premises of (RP') containing positive side formulas $L_{k+1}, \dots, L_n$, contain as well their complement, i.e., can be obtained from axioms in one step. So one can assume that the only positive atomic members in the conclusion of any ($\vee\Rightarrow$)-inference are initial clauses. Now we can write $\vee\Rightarrow$ in the form

$$\frac{\sim L_1, X_1', \sim D_1 \Rightarrow; \dots, \sim L_k, X_k', \sim D_k \Rightarrow; L_{k+1}, \sim L_{k+1} \Rightarrow; \dots; L_n, \sim L_n \Rightarrow}{\sim L_1 \vee \cdots \vee \sim L_k \vee L_{k+1} \vee \cdots \vee L_n, X', \sim D \Rightarrow}$$

so (RP') is of the form $\sim L_1 \vee \cdots \vee \sim L_k \vee L_{k+1} \vee \cdots \vee L_n; L_1 \vee D_1; \dots; L_k \vee D_k; L_{k+1} \vee \sim L_{k+1}; \dots; L_n \vee \sim L_n / D$ and dropping the last $n-k$ tautological premises we have the derivation by hyperresolution, in which all positive literals in the nucleus are preserved. Lifting in the passage from GR$'$(d) to GR(d) does not change this property, and this concludes the proof.

§**2. Modal logic S4.** In this and the following section we extend to modal logic material from the Part I, i.e., [Mints, 17]. We begin with quantified S4, i.e., with the result of adding to propositional S4 the usual quantifier postulates, which correspond to the semantics of growing domains. It is difficult to expect that our methods will be applicable to systems with the Barcan formula (except S5), since no cutfree Gentzen-type formulation is known for them.

2.1. *Modal clauses.* We again employ depth-reducing by introduction of new predicate variables to transform any formula into clause form using the equivalence

$$\forall\Box(A \leftrightarrow B) \rightarrow (F[A] \leftrightarrow F[B]) \tag{1}$$

which holds in S4 and its extensions. In fact it is sufficient to write $\forall$ only for variables free in A, B but bound in F.

We define *predicate literals* as atoms and their negations and denote them by $l, l_1, \dots$. *Modal literals* are by definition expressions of the form $l, \Box l, \Diamond l$. They are denoted by $L, M, N, L_1, M_1, N_1, \dots$. Complements are defined by $\sim\Box l = \Diamond\sim l$, $\sim\Diamond l = \Box\sim l$ in a natural way.

Predicate clauses are disjunctions of predicate literals. *Modal clauses* (or simply clauses) are disjunctions of modal literals. *Initial* modal clauses are expressions of the form $\Box\forall C$ or $\Box\forall(\exists y)C$ where C is a modal clause or has a form $\Box D$ where D is a disjunction of predicate literals. To simplify notation we require that D contain at least two terms, but this is as inessential as in Section 1.1.

We proceed as in Section 2.1.

THEOREM 2.1. *Let S be an extension of the system S4. Then for any formula F one can construct (by introduction of new variables) the list X_F of initial clauses and a propositional variable g such that*

$$\vdash_S F \text{ iff } \vdash_S \&\forall X_F \to g$$

Proof. Exactly as in the proof of Theorem 1.1 introduce predicates P_A for non-atomic subformulas A and write clauses C_A obtained from clauses in (2) Section 1.1 by prefixing $\Box\forall\boldsymbol{x}$. For example if $A = (B \;\&\; D)$ with non-atomic A, B, we put

$$\begin{aligned} C_A^+ &\equiv \{\Box\forall \boldsymbol{y}(\sim P_B(\boldsymbol{y}) \vee \sim P_D(\boldsymbol{y}) \vee P_A(\boldsymbol{y}))\} \\ C_A^- &\equiv \{\Box\forall \boldsymbol{y}(\sim P_A(\boldsymbol{y}) \vee P_B(\boldsymbol{y})), \Box\forall \boldsymbol{y}(\sim P_A(\boldsymbol{y}) \vee P_D(\boldsymbol{y}))\} \end{aligned} \tag{$2_\&$}$$

The definition is extended to $\Box$-case. If $A \equiv \Box B$ then we put

$$\begin{aligned} C_A^+ &\equiv \{\Box\forall \boldsymbol{y}(\sim P_B(\boldsymbol{y}) \vee \Box P_A(\boldsymbol{y}))\} \\ C_A^- &\equiv \{\Box\forall \boldsymbol{y}(\Diamond \sim P_A(\boldsymbol{y}) \vee P_B(\boldsymbol{y}))\} \end{aligned} \tag{$2_\Box$}$$

After this it remains only to repeat the proof of Theorem 1.1.

Since our modal systems are based on classical logic, it is easy to reduce derivability of an arbitrary formula to inconsistency of a set of clauses, i.e., to derivability of the constant $\emptyset$ or empty clause.

COROLLARY 2.2. *Under the assumptions of Theorem 2.1 provability of a formula F can be reduced to inconsistency of a set of clauses.*

Proof. Take X_F, g as in Theorem 2.1 and put $X'_F = X_F \cup \{\sim g\}$.

Note. Further simplification is possible when additional reduction axioms for modality are available. For example in S5 it is possible to consider only initial clauses of the forms:

$$\Box\forall(l_1 \vee \cdots \vee l_m)\ (m \le 3);\ \Box\forall(\exists y)(l_1 \vee l_2);\ \forall(L_1 \vee L_2) \tag{2'}$$

Indeed clauses corresponding to propositional connectives have the first of the above forms, the quantifiers add the second of these forms and clauses $(2_\Box)$ are equivalent in S5 respectively to

$$\forall \boldsymbol{y}(\Box \sim P_B(\boldsymbol{y}) \vee \Box P_A(\boldsymbol{y})) \quad \text{and} \quad \forall \boldsymbol{y}(\Diamond \sim P_A(\boldsymbol{y}) \vee \Box P_B(\boldsymbol{y})).$$

2.2. Gentzen-type modal calculus GS4. *Sequents* are expressions of the form $X \Rightarrow Y$ where X, Y are (possibly empty) lists of formulas (in the language

$\Box$, $\vee$, $\sim$). Axioms and inference rules for classical connectives and quantifiers are the same as in GK (cf. Section 1.2).

Modal rules have the following form:

$$\frac{X, (\Box A)^0, A \Rightarrow Y}{X, \Box A \Rightarrow Y}\ (\Box \Rightarrow) \qquad \frac{\Box X \Rightarrow A}{Y, \Box X \Rightarrow \Box A, Z}\ (\Rightarrow \Box)$$

Pruned derivation is again one containing thinnings only immediately preceding the last sequent of the derivation. The proof of part (a) of the following statement can be found in [Curry, 2]; the proof of (b) is standard.

THEOREM 2.3. *(a) Formula F is derivable in S4 iff the sequent $\Rightarrow F$ is derivable in GS4.*

(b) Any provable sequent has a p-inverted pruned derivation.

2.3. *Resolution calculus RS4.* Derivable objects of this calculus are modal clauses

$$L_1 \vee \cdots \vee L_p \tag{3}$$

where $L_1 \vee \cdots \vee L_p$ are modal literals, and we are interested in the derivability relations $X \vdash C$ where X is a set of initial clauses and C is a modal clause. Clauses from X are input clauses.

Axioms are initial clauses as well as $L \vee \sim L$ for modal literals L with the obvious purity restriction.

There are five inference rules. The rules (RP), $(RP_\exists)$ are as in Section 1.4, but the *nucleus* $(\exists y)^0 L_1 \vee \cdots \vee L_n$ of the rules (RP), $(RP_\exists)$ should be one of the input clauses or the result of deleting $\Box\forall$ from it. The rule

$$\frac{\Box D}{D}\ (\Box^-)$$

is to be applied only together with RP, $RP_\exists$.

Various modal systems will differ mainly by additional rules for modalities. These rules play the role somewhat similar to unification for the predicate logic. The rules for S4 are the following:

$$\frac{l \vee D}{\Diamond l \vee D}\ (\Diamond) \qquad \frac{l \vee \Diamond D}{\Box l \vee \Diamond D}\ (\Box)$$

Note that all rules are obviously valid for derivability from $\Box$-formulas in S4.

2.4. Intertranslations between Gentzen-type and resolution systems. The translation GR into GS4-derivations is defined for pruned Gentzen-type derivation $d : X \Rightarrow g$ where X is a list of initial clauses and g is a propositional variable. We extend the definition from Section 1.3. Note that any sequent in d has the form

$$X', Y \Rightarrow Y' \tag{4}$$

where X' consist of the clauses in X and the results of deleting from them some initial occurrences of $\Box, \forall$, i.e., of clauses $\Box^0 D \equiv \Box^0(L_1 \vee \cdots \vee L_n)$, $n \geq 2$. Y

is the set of remaining antecedent members. Y and Y' are lists of literals of the form $\Box l, l$. We obtain GR(d) by replacing sequents (4) by

$$\sim Y \vee Y' \tag{5}$$

and adding necessary input clauses to construct correct inferences by RP, $RP_{\exists}$.

Proceed by induction on d. Axioms are replaced as in Section 1.3. Consider the last inference L in d. If L is $\vee \Rightarrow$ apply the same transformation as in the classical case, adding ($\Box^-$) when necessary.

Let L be ($\Box \Rightarrow$). If the main formula belongs to X', simply ignore the rule. If it belongs to Y, i.e., is of the form $\Box l$, then L is transformed into the rule ($\Diamond$) of RS4: $\sim l \vee \sim Y \vee Y' / \Diamond \sim l \vee \sim Y \vee Y'$. If L is ($\Rightarrow \Box$), then it is transformed into the rule ($\Box$): $\Diamond \sim Y \vee l / \Diamond \sim Y \vee \Box l$.

The definition of the transformation GR is concluded.

The definition of the transformation RG from a derivation in RS4, $d : X_F \vdash g$, into the derivation in GS4 is modeled after Section 1.4.

We define for $d : X \vdash D$ and a given representation of D as $D_1 \vee D_2$ (modulo permutation of disjunctive members) the derivation RG$(d, D_1, D_2) : X', \sim D_1 \Rightarrow D_2$ where $\sim D_1$ consists of complements of literals in D_1.

Definition of RG(d) is given by induction on d. The main differences from Section 1.4 are in the modal rules, and we treat only them. Let L be the last inference of d. We proceed as in Part I.

Let L be $\Box^- : \Box D / D$. Here we could use the fact that $\Box D \Rightarrow D$ is derivable by ($\Box \Rightarrow$). This introduces cut, so we proceed slightly more cautiously. If $\Box D$ is initial, we obviously have $D, \sim D_1 \Rightarrow D_2$, and use $\Box \Rightarrow$. If $\Box D$ is not initial, then D is a literal, and dropping $\Box$ from all predecessors of $\Box D$ in a given Gentzen-type derivation of $X \Rightarrow \Box D$ we have a derivation of $X \Rightarrow D$ where some axioms $C \vee \Box D \Rightarrow C \vee \Box D$ are replaced by $C \vee \Box D \Rightarrow C \vee D$, but these are easily derivable.

Let L be ($\Diamond$): $l \vee C \,/\, \Diamond l \vee C$. Then it is transformed into $\Rightarrow \Diamond$ if l is in D_2, or into $\Box \Rightarrow$ if l is in D_1 according to given partition $\Diamond l \vee C = D_1 \vee D_2$.

Let L be ($\Box$). It is transformed into $\Rightarrow \Box$ or $\Diamond \Rightarrow$. Proviso for antecedent members is satisfied, since all initial clauses begin with $\Box$.

The description of RG(d) is finished.

THEOREM 2.4. (Soundness and Completeness Theorem.) *Let F be a modal formula, and X_F is as in Theorem 2.1.*

(a) If $d : X_F \Rightarrow g$ is the derivation in LS4, then $GR(d) : X'_F \vdash g$ (or $X_F \vdash \emptyset$) is a derivation in RS4, where X' is a sublist of X_F.

(b) If $d : X_F \vdash g$ (or $X_F \vdash \emptyset$) in RS4 then $RG(d) : X_F \Rightarrow g$ is the derivation in GS4.

(c) $\vdash_{\mathrm{S4}} F$ iff $X_F \vdash_{\mathrm{RS4}} g$.

Proof. (a), (b) were established during the definitions of GR, RG, and (c) follows from (a), (b), and Theorem 2.1. Q.E.D.

§3. Modal logic S5. Since cutfree Gentzen-type formulations are known for the quantified systems T, K4, and K, there seems to be no difficulty in extending to them the formulations and results of Section 4 along the lines of Part I (more precisely Section 4 of [Mints, 17]). For the quantified S5 the situation is different from the propositional case, where there exists a cutfree formulation complete for modalized formulas [Shvartz, 18] which was used in Part I, as well as a formulation with analytic cut complete for all formulas. The best existing approximation to a cutfree system is the formulation in terms of systems of sequents (semantical tableaux) due to Kripke [6] and Kanger [3]. The formulation of Mints [12] in terms of systems of sequents is essentially equivalent. We present here the modification of our approach suitable for this situation. The general schema is as before: the resolution derivation is obtained by moving the atomic part of the Gentzen-like derivation in the succedent, but now the original objects are more complicated, and this will be reflected in the more complex structure of clauses.

1. System TS5 of semantic tableaux for S5. We describe a system TS5 which is similar to system LS5 in Mints [12]. The main difference is that the sequents will now consist not only of a succedent, as in LS5.

Let a *tableau* be any expression of the form $\{S\}$ where S is a sequent. The expression $\{\ \}$ is treated as the constant *false*. Capital Greek letters Γ, Π, Φ etc. stand for sequents.

Arbitrary lists of tableaux are called systems (of tableaux) and denoted by S, T, U, V etc. We disregard the order of tableaux in a system.

The non-modal postulates (i.e., axioms and inference rules) of the system TS5 will be essentially the same as in GK. More precisely they will be obtained from the corresponding postulates of GK by adding arbitrary tableaux. Modal rules correspond to the Kripke semantics of S5-modality.

We shall ignore the order of members in a tableau and the order of tableaux in a system.

The translation of a tableau $\{A_1, \ldots, A_n \Rightarrow B_1, \ldots, B_m\}$ is a formula $\Box({\sim}A_1 \vee \cdots \vee {\sim}A_n \vee B_1 \vee \cdots \vee B_m)$. *The translation of a sequent* is the disjunction of translations of its member tableaux.

Axioms: $\{A \Rightarrow A\}$

Inference rules:

$$\frac{\{X \Rightarrow Y, A_1, \ldots, A_m\}, S}{\{X \Rightarrow Y, A_1 \vee \cdots \vee A_n\}, S} \begin{matrix}(\Rightarrow\vee)\\ m \le n\end{matrix} \qquad \frac{\{X \Rightarrow Y\}, S}{\{X', X \Rightarrow Y, Y'\}, S, S'} \text{(thinning)}$$

$$(\Rightarrow{\sim})\frac{\{A, X \Rightarrow Y\}, S}{\{X \Rightarrow Y, {\sim}A\}, S} \qquad \frac{\{X \Rightarrow Y, A\}, S}{\{{\sim}A, X \Rightarrow Y\}, S}({\sim}\Rightarrow)$$

$$\frac{\{A_1, X_1 \Rightarrow Y_1\}, S_1; \ldots; \{A_n, X_n \Rightarrow Y_n\}, S_n}{\{A_1 \vee \cdots \vee A_n, X \Rightarrow Y\}, S} (\vee\Rightarrow)$$

where $X_1 \cup \cdots \cup X_n = X$, $Y_1 \cup \cdots \cup Y_n = Y$ and $S_1 \cup \cdots \cup S_n = S$, i.e., each of the tableaux S_i is obtained from S by deleting whole tableaux and/or members of tableaux, and each formula in S is retained in at least one of the S_i.

$$(\forall\Rightarrow)\ \frac{\{A[x := t], (\forall x A)^0, X \Rightarrow Y\}, S}{\{\forall x A, X \Rightarrow Y\}, S} \qquad \frac{\{X \Rightarrow Y, A[x := b]\}, S}{\{X \Rightarrow Y, \forall x A\}, S}\ (\Rightarrow\forall)$$

$$(\exists \Rightarrow)\ \frac{\{A[x:=b], X \Rightarrow Y\}, S}{\{(\exists x)A, X \Rightarrow Y\}, S} \qquad \frac{\{X \Rightarrow Y, ((\exists x)A)^0, A[x:=t]\}, S}{\{X \Rightarrow Y, (\exists x)A\}, S}\ (\Rightarrow \exists)$$

with usual proviso for $(\Rightarrow \forall), (\exists \Rightarrow)$: the eigenvariable b does not occur free in the conclusion.

$$\frac{\{M \Rightarrow\}, \{X \Rightarrow y\}, S}{\{M, X \Rightarrow Y\}, S} \quad (M) \quad \frac{\{\Rightarrow M\}, \{X \Rightarrow Y\}, S}{\{X \Rightarrow Y, M\}, S}$$

where M is a modalized formula.

$$(\Box \Rightarrow)\ \frac{\{\Box A \Rightarrow\}^0, \{A, X \Rightarrow Y\}, S}{\{\Box A \Rightarrow\}, \{X \Rightarrow Y\}, S} \qquad \frac{\{\Rightarrow A\}, S}{\{\Rightarrow \Box A\}, S}\ (\Rightarrow \Box)$$

$$(\Diamond \Rightarrow)\ \frac{\{A \Rightarrow\}, S}{\{\Diamond A \Rightarrow\}, S} \qquad \frac{\{\Rightarrow \Diamond A\}^0, \{X \Rightarrow Y, A\}, S}{\{\Rightarrow \Diamond A\}, \{X \Rightarrow Y\}, S}\ (\Rightarrow \Diamond)$$

Comments. During the proof search process (or in the derivation viewed bottom-up) new tableaux in the system arise only in the rule (M), but they are used in an essential way only in the rules $(\Rightarrow \Box)$, $(\Diamond \Rightarrow)$. The tableaux in the system correspond to different worlds in the Kripke model. The rule (M) says that a modalized formula has the same value in all worlds. The rules $(\Box \Rightarrow)$ and $(\Rightarrow \Diamond)$ express that if $\Box A$ is true ($\Diamond A$ is false) in some world, then A is true (false, respectively) in any world.

Example 1. Let us derive the Barcan formula.

$$\cfrac{\cfrac{\cfrac{\cfrac{\{P(a) \Rightarrow P(a)\}}{\{\Box p(a) \Rightarrow\}, \{\Rightarrow P(a)\}}}{\{\forall x \Box P(x) \Rightarrow\}, \{\Rightarrow P(a)\}}}{\{\forall x \Box P(x) \Rightarrow\}, \{\forall x P(x)\}}}{\{\forall x \Box P(x) \Rightarrow \Box \forall x P(x)\}}$$

The main step is the third one, where (if we view it bottom-up) the individual a from the second world appeared in the first world, i.e., the symmetry of the accessibility relation between worlds was implicitly used.

Equivalence of TS5 to more familiar formulations is easily established by reference to [Kripke, 6] or [Mints, 12], and we shall not go into details of this.

3.2. Resolution calculus RS5. According to Note 1 in Section 2.1 each formula F can be reduced in S5 to the sequent $X_F \Rightarrow g$ where X_F is a list of clauses of the form 2.1(2′). Using the Barcan formula to interchange $\Box$ and $\forall$, and dropping initial $\forall$ we can put them into the form:

$$(l_1 \vee \cdots \vee l_m), m \leq 3; \qquad (\exists y)(l_1 \vee l_2); \qquad L_1 \vee L_2 \tag{1}$$

where L_1, L_2 are modal literals containing $\Box$ or $\Diamond$.

Let us call (1) *initial clauses*, and define *modal clauses* (for S5) to be disjunctions

$$\Box D_1 \vee \cdots \vee \Box D_n \vee \Diamond D_{n+1} \tag{2}$$

where each of the D_i is the disjunction of predicate (non-modal) literals and $\Diamond(l_1 \vee \cdots \vee l_k)$ is understood as $\Diamond l_1 \vee \cdots \vee \Diamond l_k$ and $\Box\emptyset \equiv \Diamond\emptyset \equiv \emptyset$. We disregard as before the order and repetitions of terms in disjunctions. Note that these objects are more complicated than modal clauses for S4 as defined in Section 3.

Let us describe a resolution system for relations $X \vdash C$, i.e., for deriving modal clauses C from the set X of initial clauses. The clauses from X are input clauses.

Resolution system RS5.

Axioms: Input clauses, $\Box(l \vee \sim l)$, $L \vee \sim L$ for modalized L with purity restriction.

Inference rules are (RP), $(RP_\exists)$ in each disjunctive member of (2) and S5-modal rule. More precisely:

$$(RP) \qquad \frac{\Box(l_1 \vee \cdots \vee l_n); \Box(\sim l_1' \vee D_1) \vee \underline{D}_1; \Box(\sim l_n' \vee D_n) \vee \underline{D}_n}{\Box(D_1 \vee \cdots \vee D_n)\sigma \vee \underline{D}}$$

where $\sigma \equiv \mathrm{MGU}(l_1, l_1'; \ldots; l_n, l_n')$, $\underline{D} \equiv \underline{D}_1\sigma \cup \cdots \cup \underline{D}_n\sigma$, as well as a $(L_1 \vee L_2)$-version:

$$(RP) \qquad \frac{L_1 \vee L_2; \sim L_1' \vee D_1; \sim L_2' \vee D_2}{(D_1 \vee D_2)\sigma}$$

$$(RP_\exists) \qquad \frac{\Box(\exists y)(l_1 \vee l_2); \Box(\sim l_1' \vee D_1) \vee \underline{D}_1; \Box(\sim l_2' \vee D_2) \vee \underline{D}_2}{\Box(D_1 \vee D_2)\sigma \vee \underline{D}}$$

where $\sigma \equiv \mathrm{MGU}_\exists(l_1, l_1'; \ldots; l_n, l_n')$, $\underline{D} = D_1\sigma \cup D_2\sigma$.

$$(\Box \rightarrow \Diamond) \; \frac{\Box(l \vee D) \vee \underline{D}}{\Diamond l \vee \Box D \vee \underline{D}}$$

Factorization rule (F) is as usual.

Example 2. $p \rightarrow \Box\Diamond p$.

Introducing variables x for $\Diamond p$, y for $\Box x$, z for $p \rightarrow y$ and using reduction to a clause form taking signs into account, we have the problem:

$$(1)\, \Box{\sim}p \vee \Box x, (2)\Diamond{\sim}x \vee \Box y, (3)\Box({\sim}y \vee z), (4)\Box(p \vee z) \vdash \Box z$$

The derivation (by semi-input resolution is as follows.

$$\cfrac{\Box({\sim}y \vee z) \quad \cfrac{\Box(p \vee z) \quad \cfrac{\Box{\sim}p \vee \Box x \quad \Diamond{\sim}x \vee \Box y}{\Box{\sim}p \vee \Box y}}{\Box z \vee \Box y}}{\Box z \vee \Box z \equiv \Box z}$$

Example 3. Barcan formula. Introducing the variable p for $\forall x Px$ we have the problem

$$\Box(\exists y)({\sim}P(y) \vee p), \Box P(x) \vdash p$$

which is solved in one step of $(RP_\exists)$.

The description of the algorithm GR translating TS5-derivations into RS5-derivations is by now standard. Delete from the given derivation all modalized formulas which are not literals, move modalized literals into separate tables and replace each system

$$\{X_1 \rightarrow Z_1\}, \ldots, \{X_n \rightarrow Z_n\}$$

by the clause

$$\Box(\sim X_1 \vee Z_1) \vee \cdots \vee \Box(\sim X_n \vee Z_n)$$

where $\Box$ is not prefixed if a table consists of the only modalized formula.

Then the rule ($\vee \Rightarrow$) is transformed into (a modal version of) (RP') or ($RP'_\exists$) (cf. Section 1.3) depending on the initial clause which is an ancestor of the main formula of the ($\vee \Rightarrow$) considered. The rules ($\Box \Rightarrow$), ($\Rightarrow \Diamond$) having a modal literal as main formula are transformed into ($\Box \rightarrow \Diamond$). After this, lifting is applied as in Section 1.3 to assure the standard form of the rules.

Combined with soundness of the rules for RS5 this establishes the following.

THEOREM 3.1. *The system RS5 is complete for S5-derivability of relations* $\forall X \vdash C$ *where* X *is a list of initial clauses and* C *is a modal clause (modulo subsumption).*

REFERENCES

[1] C.-L. CHANG and R. LEE. ***Symbolic logic and mechanical theorem proving.*** Academic Press, New York, 1973.

[2] H. CURRY. ***Foundations of mathematical logic.*** McGraw-Hill, New York, 1963.

[3] S. KANGER. ***Handbook i logic.*** (Mimeographed.) Stockholm, 1959.

[4] S. C. KLEENE. ***Introduction to metamathematics.*** Van Nostrand, Princeton, 1952; reprinted by North-Holland, Amsterdam.

[5] S. C. KLEENE. *Permutability of inferences in Gentzen's calculi LK and LJ.* ***Two papers on the predicate calculus***, Memoirs of the American Mathematical Society, vol. 10, 1952.

[6] S. KRIPKE. *A completeness theorem in modal logic.* ***The journal of symbolic logic***, vol. 24 (1959), pp. 1–14.

[7] V. LIFSCHITZ. *What is the inverse method?* ***Journal of automated reasoning***, vol. 5 (1989), pp. 1–23.

[8] S. YU. MASLOV. *The inverse method of establishing deducibility* (Russian). ***Trudy Matematicheskogo Instituta imeni V. A. Steklova***, vol. 98 (1968), pp. 26–87. (Translated by the American Mathematical Society.)

[9] S. YU. MASLOV. *Proof search strategies based on the ordering in a favorable set.* ***Seminars in mathematics***, vol. 16, Plenum Press, New York, 1971.

[10] S. YU. MASLOV. *Connection between the strategies of the inverse method and the resolution method.* ***Seminars in mathematics***, vol. 16, Plenum Press, New York, 1971.

[11] S. YU. MASLOV. ***Theory of deductive systems and its applications.*** MIT Press, Cambridge, Mass, 1987.

[12] G. MINTS. *Lewis systems and the system T* (Russian). In: R. Feys,

Modal logic, Nauka, Moscow, 1974. English translation in G. Mints, ***Proof-theoretic transformations***, Bibliopolis, Napoli, 1990.

[13] G. MINTS. *Resolution calculi for the non-classical logics* (Russian). ***9th Soviet Symposium in Cybernetics***, VINITI, Moscow, 1981.

[14] G. MINTS. *Resolution calculi for the non-classical logics* (Russian). ***Semiotics and informatics***, vol. 25 (1985), pp. 120–135.

[15] G. MINTS. *Resolution calculi for modal logics* (Russian). ***Proceedings of the Estonian Academy of Science***, N3 (1986), pp. 279–290. (English translation in ***Translations of the AMS***, series 2, vol. 143, pp. 1–10, 1989.)

[16] G. MINTS. *Cutfree formalisations and resolution methods for propositional modal logic.* ***VIII International Congress for Logic, Methodology and Philosophy of Science***, Moscow, 1987, pp. 46–48.

[17] G. MINTS. *Gentzen-type systems and resolution rules. Part I. Propositional logic.* ***COLOG-88***, Lecture notes in computer science, vol. 417, Springer-Verlag, Berlin, Heidelberg, New York, 1990, pp. 198–231.

[18] G. SHVARTZ. *Gentzen style systems for K45 and K45D.* Lecture notes in computer science, vol. 363, Springer-Verlag, Berlin, Heidelberg, New York, 1989, pp. 245–256.

[19] N. ZAMOV. *Resolution without Skolemization.* ***Doklady Akademii Nauk SSSR***, vol. 293, N5 (1987), pp. 1046–1049.

[20] N. ZAMOV. *Maslov's inverse method and decidable classes.* ***Annals of pure and applied logic***, vol. 42 (1989), pp. 165–194.

Institute of Cybernetics
Estonian Academy of Sciences
Tallinn 200108
Estonia