Intercalation Calculi for Classical Logic

by

Wilfried Sieg

January 1994

Report CMU-PHIL-46

Carnegie Mellon

Philosophy
Methodology
Logic

Pittsburgh, Pennsylvania 15213-3890

INTERCALATION CALCULI FOR CLASSICAL LOGIC*

Wilfried Sieg
Department of Philosophy
Carnegie Mellon University
Pittsburgh, PA 15213

*This material is partially based upon work supported by the National Science Foundation under Grant No. CCR-9206756. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

ABSTRACT. Natural deduction (for short: nd-) calculi have not been used systematically as a basis for automated theorem proving in classical logic. To remove objective obstacles to their use I describe (1) a method that allows to give *semantic proofs of normal form theorems for nd-calculi* and (2) a framework that allows to *search directly for normal nd-proofs*. Thus one can try to answer the question: How do we bridge the gap between claims and assumptions in heuristically motivated ways? This informal question motivated the formulation of *intercalation calculi*. Ic-calculi are the technical underpinnings for (1) and (2), and the paper focuses on their detailed presentation and meta-mathematical investigation -- in the case of classical predicate logic.[1]

1. Search for proofs. Natural deduction calculi have been available since the mid-thirties and reflect "as accurately as possible the actual logical reasoning involved in mathematical proofs".[2] They capture the logical structure of arguments by incorporating *inferences from* and *to* complex formulas with characteristic principal connective. The rules for the logical connectives, in the case of sentential logic $\wedge$, $\vee$, $\rightarrow$, and $\neg$, are consequently divided into "Elimination", i.e. E-, and "Introduction", i.e. I-rules. (The considerations will be extended to predicate logic in section 6.) I give only the rules for negation, because they are formulated here in a way that is not in the standard (Gentzen-Prawitz) mould. The negation elimination rule $\neg$E is the distinctive rule of classical logic and is needed, for example, to prove the law of excluded middle and Peirce's law; the introduction rule $\neg$I captures the form of indirect argumentation as used in the Pythagorean proof of the irrationality of $\sqrt{2}$.

$$\frac{\begin{array}{cc}[\neg\phi] & [\neg\phi]\\ \vdots & \vdots \\ \varphi & \neg\varphi\end{array}}{\phi} \qquad\qquad \frac{\begin{array}{cc}[\phi] & [\phi]\\ \vdots & \vdots \\ \varphi & \neg\varphi\end{array}}{\neg\phi}$$

Generally, E-rules specify how components of assumed or established complex formulas can be used in an argument; I-rules provide conditions under which complex formulas can be inferred from already established components. This leads directly to the formulation of very intuitive strategies.

---

[1] The work reported here continues the metamathematical investigations basic for the Carnegie Mellon Proof Tutor, see [Sieg and Scheines]. The motivation for formulating ic-calculi comes directly from our joint work on automated proof search; cf. notes 7 and 8.

[2] Gentzen in his "Investigations into logical deduction", [Szabo], p. 74.

The strategies exploit, that the structure of nd-proof can be made to depend crucially on the *syntactic context* provided by assumptions and conclusions: The nd-calculi share, as Prawitz [1965] discovered, important metamathematical properties with sequent calculi. For the statement of the first of these properties recall that the premise of an E-rule with the characteristic connective is called *major premise;* a proof is called *normal,* when, roughly speaking, no formula occurrence in the proof is both the conclusion of an I-rule (or of ¬E) and the major premise of an E-rule. The *first central property* was established by Prawitz using a special sequence of "reductions" to transform arbitrary proofs into normal ones:

Normalization Theorem.[3] Any proof of G from α in the nd-calculus can be transformed into a normal proof leading from α to G.

Here α is a sequence of formulas. The *second central property* for the nd-calculus is a structural property of certain proofs: normal proofs D leading from α to G have the *subformula property;* i.e., every formula occurring in D is (the negation of) either a subformula of G or of an element in α.

Despite the naturalness of nd-calculi, the part of proof theory that deals with them has hardly influenced developments in automated theorem proving. For that the proof theoretic tradition rooted in Herbrand's work and Gentzen's work on sequent calculi has been more important. The keywords here are *resolution* and *logic programming.* From a purely logical point of view this is prima facie peculiar: It is after all the subformula property of special kinds of derivations[4] that makes resolution and related techniques possible; and normal derivations in natural deduction calculi have that very property (with the minor addition mentioned above). Why is it then that nd-calculi have not been exploited for **automated proof search**? The answer to this general question lies, it seems to me, in answers to **three crucial questions:** **(1)** How can one specify through a calculus *exactly* the normal proofs? **(2)** How can one construct a search space that allows the formulation of strategies for finding such proofs? and **(3)** How can one prove the termination of search strategies?

---

[3] Prawitz formulated the theorem only for a part of the classical calculus. The (strong) normalization theorem for the full calculus was established by Stalmark (1991).

[4] Derivations in Herbrand's calculus and derivations in the sequent calculus without cut have the *subformula property*: they contain only subformulas of their endformula, respectively endsequent.

In the *case of sequent calculi* the analogues to these questions have direct answers: Use calculi without the cut rule; invert systematically their rules; prove their completeness! In this rough description of automated deduction based on sequent calculi the syntactic normalization or cut-elimination procedure is not mentioned, since the semantic completeness proof for the cut-free part is fundamental; algorithms for finding *cut-free* derivations are refinements of strategies used in that proof. Such strategies **realize the heuristic idea of *searching for semantic counterexamples* and yield** trees a such that *either* one of a's branches allows the definition of a counter-example to "a has G as a logical consequence" *or* a constitutes a cut-free derivation of the sequent -i(x,G.[5] In the *case of nd-calculi* normal proofs are also sufficient to obtain all logical consequences from given assumptions. However, this fact has not been established directly: Its proof combines the completeness theorem for the calculus with the normalization theorem. In order to obtain a direct proof of the fact and an answer to (1) *intercalation calculi* are introduced. They provide frameworks for answering (2), and completeness proofs for these calculi answer (3).

2. *Ic-calculi (for sentential logic).* The broad problem is this: How can we derive a conclusion or goal G from assumptions $\phi_1, \ldots, \phi_n$? Or, more vividly: How can we close via logical rules the gap between G and the $\phi_1, \ldots, \phi_n$? This question is at the heart of spanning search spaces via ic-calculi. The basic rules of such calculi are reformulations of those for Gentzen's nd-calculi, but it is the preservation of inferential information and the restricted way in which the rules are used to close the gap (and thus to build up derivations) that is distinctive. I will discuss at first only classical sentential logic; however, the considerations will then be extended to predicate logic and can be used to treat non-classical logics, for example, intuitionistic logic.[6]

The intercalation rules operate on triples of the form a;P?G. a is the sequence of *available assumptions;* G is the current *goal;* |3 is a sequence of formulas obtained by A-elimination and -»-elimination from elements in a. To facilitate the description of rules and parts of search trees let us agree on

^ -HI consists of the negated formulas in a.

[6] That was done for sentential logic by Cittadini in his M.S. thesis written in May 1991; see [Cittadini 1992]. The case of intuitionistic predicate logic and other non-classical logics will be considered in a joint paper with Cittadini, "Intercalation calculi for non-classical logics".

some conventions. Lower case Greek letters α, β, γ, δ, ... range over finite sequences of formulas; as syntactic variables over formulas I use ϕ, ψ, χ, ...; ρ, σ, τ, ... range over trees. At first I consider only formulas in the language of sentential logic with the connectives $\neg$, $\wedge$, $\vee$, $\rightarrow$; ($\bot$ will be used as an auxiliary symbol). $\phi\in\alpha$ expresses that ϕ is an element of the sequence α; α,β is short for the concatenation $\alpha*\beta$ of the sequences α and β; α,ϕ stands for the sequence $\alpha*\langle\phi\rangle$, where $\langle\phi\rangle$ is the sequence with ϕ as its only element. There are three kinds of ic-rules: those corresponding to E-rules for $\wedge$, $\vee$, $\rightarrow$; those corresponding to I-rules for $\wedge$, $\vee$, $\rightarrow$; finally, rules for negation. Let me list the rules of the first kind, i.e. $\downarrow$-rules.

$\downarrow\wedge_i$: $\alpha;\beta?G,\ \phi_1\wedge\phi_2\in\alpha\beta,\ \phi_i\notin\alpha\beta \Rightarrow \alpha;\beta,\phi_i?G$ for i=1 or 2

$\downarrow\vee$: $\alpha;\beta?G,\ \phi_1\vee\phi_2\in\alpha\beta,\ \phi_1\notin\alpha\beta,\ \phi_2\notin\alpha\beta \Rightarrow \alpha,\phi_1;\beta?G$ AND $\alpha,\phi_2;\beta?G$

$\downarrow\rightarrow$: $\alpha;\beta?G,\ \phi_1\rightarrow\phi_2\in\alpha\beta,\ \phi_2\notin\alpha\beta,\ \phi_1\neq G \Rightarrow \alpha;\beta?\phi_1$ AND $\alpha;\beta,\phi_2?G$

The side conditions of these rules help us to avoid "repeating questions": $\alpha;\beta?G$ is the *same question as* $\alpha^*;\beta^*?G$ just in case the sets of formulas in the sequences α,β and α^*,β^* are identical. Now I formulate the rules that correspond to inverted introduction rules, i.e. $\uparrow$-rules.

$\uparrow\wedge$: $\alpha;\beta?\phi_1\wedge\phi_2 \Rightarrow \alpha;\beta?\phi_1$ AND $\alpha;\beta?\phi_2$

$\uparrow\vee_i$: $\alpha;\beta?\phi_1\vee\phi_2 \Rightarrow \alpha;\beta?\phi_i$ for i=1 or 2

$\uparrow\rightarrow$: $\alpha;\beta?\phi_1\rightarrow\phi_2 \Rightarrow \alpha,\phi_1;\beta?\phi_2$

The rules for negation are split into three, where $\bot$ is considered as a place-holder for (the conjunction of) a pair of contradictory formulas; this purely auxiliary role of $\bot$ accounts for the restriction $\phi\neq\bot$ in the first rule:

$\bot_c$: $\alpha;\beta?\phi,\ \phi\neq\bot \Rightarrow \alpha,\neg\phi;\beta?\bot$

$\bot_i$: $\alpha;\beta?\neg\phi \Rightarrow \alpha,\phi;\beta?\bot$

$\bot_{\mathfrak{F}}$: $\alpha;\beta?\bot,\ \varphi\in\mathfrak{F}(\alpha) \Rightarrow \alpha;\beta?\varphi$ AND $\alpha;\beta?\neg\varphi$.

In the last rule F(a) is the class of all PROPER subformulas of elements in a; clearly, *L?* is inapplicable in case *T*(a) is empty. F(a) is always finite; and that is crucial for the finiteness of the search space. Operations 0 leading to smaller and yet sufficient classes can be specified; cf. section 5. The different calculi we are considering are distinguished through the operation 0, and I denote a particular calculus by IC(O).

Remark: Intuitionistic versions of ic-calculi are obtained by using the rule cc;p?<|), <t>/\l => a;p?l *(ex falso quodlibet)* instead of l_c. For the classical systems *-> can be weakened to: a;p?G, <|>i-H>2€ap, <l>i€aP> <|>2$aP => a;P,<t>2?G; but this formulation, as Cittadini noticed, is too weak for intuitionistic logic (and unnatural for proof search in the classical case).

3. *The problem space.* The ic-calculus provides the underpinning for specifying informal approaches to proof search: Its rules are used to construct a search space that contains all possible ways of closing the gap between a and G via the ic-rules. In this space we search for a gap-closing subtree that determines, in turn, a unique normal nd-derivation from a to G; if the search fails, the search space contains enough information to yield a semantic counterexample. This sketch of the completeness proof for the ic-calculus shows the family resemblance to completeness proofs for the sequent calculus without cut. The difference can be put sharply as follows: *In the case of the* ***sequent calculus, one tries to find a semantic counterexample and, if that search fails, one actually has found a proof; in the case of the ic-calculus, one tries to find a proof and, if that search fails, one has a counterexample.***

As an example of how the intercalation rules are used to build up the search space for a question a;?G, let me show the search tree for the question ?Pv-iP. It is *partially* presented in Diagram 1 of the Appendix. We start out by applying three intercalation rules to obtain three new questions, namely, ?P OR ?-.P OR, proceeding indirectly, -•(Pv--P);?l. That the branching in the tree is disjunctive is indicated by D. Let us pursue the leftmost branch in the tree: To answer ?P we have to use l_c and, because of the restriction on the choice of contradictory pairs, we have only to ask -"P;?P AND -«P;?->P. B indicates

---

[7] **A sequent proof is far from reflecting the structure of ordinary arguments. Thus, we have here and in the case of resolution based procedures the non-trivial problem of finding associated nd-proofs. Cf. Shanin e.a., but also Andrews and Pfenning. The issue is also addressed in implementations of, e.g., NUPRL and ISABELLE. Bledsoe's way of using nd-methods is not systematic in the logical setting.**

that the branching is conjunctive here. In the first case only $\bot_c$ can be applied and leads to the same question we just analyzed: Using $\neg P$ as an assumption, $\bot$ has to be proved. Thus we *close this branch* with a circled N, linking it to the same earlier question on the branch. In the second case the gap between assumptions and goal is obviously closed, so we top this branch with a circled Y. The other parts of the tree are constructed in a similar manner. But the tree is not quite full: At the nodes that are distinguished by arrows the additional contradictory pair consisting of P and $\neg P$ has to be considered. At nodes 2 and 3 the resulting branches do not help in closing the gap; at node 1, in contrast, the resulting subtree is of interest.

The darkened branches of Diagram 1 contain enough information for the extraction of derivations in a variety of styles of natural deduction. For our calculus we can easily obtain corresponding derivations; namely, first:

$$\dfrac{\dfrac{\dfrac{\dfrac{\cancel{\neg P}}{P \vee \neg P} \quad \cancel{\neg(P \vee \neg P)}}{P}}{P \vee \neg P} \quad \cancel{\neg(P \vee \neg P)}}{P \vee \neg P}$$

The second derivation is analogous to this one, except that the roles of P and $\neg P$ are interchanged; finally, the derivation that emerges from the undrawn part at node 1 is this:

$$\dfrac{\dfrac{\dfrac{\cancel{\neg P}}{P \vee \neg P} \quad \cancel{\neg(P \vee \neg P)}}{P} \qquad \dfrac{\dfrac{\cancel{P}}{P \vee \neg P} \quad \cancel{\neg(P \vee \neg P)}}{\neg P}}{P \vee \neg P}$$

The (full) search or ic-tree is specified inductively by applying the ic-rules to the initial question or to the "non-terminal" leaves of an already obtained partial search tree. In either case one addresses questions of the form $\alpha^*;\beta^*?G^*$. We call a branch determined by $\alpha^*;\beta^*?G^*$ a *left→-branch*, if an earlier question on that branch is the left consequence of $\downarrow\rightarrow$, and distinguish two main cases:

1. G* is different from 1 and (i) cc*;p*?G* is not on a left-»-branch, then apply intercalation rules in all possible ways, e.g., in the order IAJ, IA_2, !-», Iv, IA, t->, tv_{1f} tv_2, and finally either Ij or l_c, **unless** G*€oc*,p*; in this case close with Y; (ii) a*;P*?G* is on a left->-branch and does not repeat an earlier question, then proceed as under (i); (iii) oc*;p*?G* is on a left-»-branch and does repeat an earlier question, then close with N.

2. G* is 1: apply ly with **q>€F(a*), unless F(a*)** is empty or there is a question ai;Pi?l on the branch determined by oc*;P*?l with F(a*)=F(oci); in these cases close with N.

The point of N-closing branches is to avoid repeating questions. Due to the side conditions for the rules, they all guarantee that the generated questions are new — except for *-» and If: in the unless-case in (2) no new assumptions have been added; consequently, all the questions that could be asked by going indirect at this point are being asked on branches emerging from the node with cci;pi?i. (That this restriction is correct and not too restrictive is of course shown by the considerations below.) — The ic-tree is constructed in this way for questions a;?G; its branches constitute sequences of *subquestions* for a;?G of the form <OCJ;PJ?GJ>J€|, where I is a subset of N. Such sequences satisfy the obvious conditions: (i) cco;po?Go *s a»?G, and (ii) for any i>0 the element a,;pj?Gj is obtained from the immediately preceding subquestion as (one of) the conclusion(s) of an ic-rule. Due to the finiteness of F, the restrictive condition for J->, and the complexity reducing character of the remaining I- and t-rules all sequences of sub-questions are finite; as the ic-tree is also finitely branching we have the first part of the following proposition:

Proposition. The ic-trees for questions a;?G are finite, and their branches are closed with either Y or N.

Proof. Because of the above observations, only the second part of the proposition has to be established. So assume that a particular leaf with question cc*;p*?G* is not closed with Y. Then we must have, first of all, that neither a l-rule, nor a t-rule is applicable; ij and l_c are also not applicable. So either we are in case (l)(iii), or G* is i! But then the construction is terminated, because a question is repeated or if is not applicable, and the branch is closed with N. Q.E.D.

Every branch in an ic-tree is finite and is topped by either a circled Y or N. This assignment to the leaves can be extended to the whole tree and determines a unique value for the original question; the value of a question a;p?G is indicated by [a;p?G]. In the next two sections I will show two facts: (1) If Y is assigned to the root of the ic-tree, then there is a normal derivation leading from the assumptions to the goal of the question; (2) If N is assigned to the root of the ic-tree, then there is not only no normal derivation, but no derivation at all: The ic-tree contains enough information to show that the inference from a to G is semantically invalid.

4. Extracting Normal Proofs. We saw through the Pv-^P-example, how an nd-proof can be read off from a properly chosen partial ic-tree whose root evaluates to Y. To formulate the underlying general fact properly I define first the notion of an *ic-derivation.*

Definition. An *ic-derivation for the question* a;p?G is a subtree x of the ic-tree a for a;p?G satisfying: (i) a;p?G is the root of x, (ii) all branches of x are Y-closed branches of a, and (iii) to every node in x (that is not a leaf) exactly one ic-rule is applied to obtain its successor(s).

One can easily extract ic-derivations from ic-trees that evaluate to Y. Let a be the intercalation tree for the question oc;?G and assume that [a;?G] = Y. We can determine from a a canonical Y-subtree x as f(dp(a)), where dp(a) is the depth of a and f a function defined recursively as follows:

f(0)	=	a;?G	
f(n+1)	=	e(f(n))	if some branches of f(n), viewed as a subtree of a, are still open
		f(n)	otherwise

e extends the open branches of the tree f(n) by their "left-most Y-expansions" in a. More explicitly, the open branches of f(n) are open branches of a and are consequently expanded by ic-rules; at least one of these rules must have a (pair of) conclusion(s) evaluating to Y; choose the left-most such rule application in each case. — The main point for sure is this: from an ic-derivation we can construct uniquely an associated nd-proof. For the proof of this fact I reformulate the rules for negation ever so slightly:

l_c: oc;p?G, cp € f(a,--G) => a,->G;p?cp AND a,-"G;p?->(p ,

1J: a;p?-iG,(p€f(a,G) => a,G;p?q> AND a,G;p?-cp.

Note also, for the sake of expositional completeness, that the corresponding reformulation of ex falso quodlibet for the intuitionistic calculus is given by: $\alpha;\beta?G,\ \varphi \in \mathbf{F}(\alpha) \Rightarrow \alpha;\beta?\varphi$ AND $\alpha;\beta?\neg\varphi$.

Proposition. For any **I**, α, β, G: if **I** is an ic-derivation for $\alpha;\beta?G$, then there is a uniquely determined normal nd-proof $\mathbf{D_I}$ leading from α,β to G.

Proof (by induction on the height of **I**). If hg(**I**)=1, the ic-derivation simply consists of the question $\alpha;\beta?G$ with $G \in \alpha,\beta$, as **I** evaluates as **Y**. $\mathbf{D_I}$ is the nd-proof consisting of the node G. -- If hg(**I**)>1, distinguish cases as to the ic-rule that is applied to $\alpha;\beta?G$ in **I**. The induction hypothesis asserts: for any ic-derivation **J** with hg(**J**)<hg(**I**) there is a uniquely determined normal nd-proof $\mathbf{D_J}$ answering the question at the root of **J**.

$\downarrow\wedge_i$: The immediate subtree $\mathbf{J_i}$ of **I** has root $\alpha;\beta,\phi_i?G$; by induction hypothesis there is a uniquely determined normal nd-proof $\mathbf{D_{J_i}}$ leading from assumptions in α,β,ϕ_i to G. If $\mathbf{D_{J_i}}$ contains any occurrences of ϕ_i as open assumptions, then replace those occurrences by

$$\frac{\phi_1 \wedge \phi_2}{\phi_i}$$

The resulting normal proof of G from α,β is the associated nd-proof $\mathbf{D_I}$.

$\downarrow$**v**: The immediate subtrees $\mathbf{J_i}$ of **I** have roots $\alpha,\phi_i;\beta?G$ for i=1 or 2; by induction hypothesis there are uniquely determined normal nd-proofs $\mathbf{D_{J_i}}$ leading from α,ϕ_i,β to G. The associated normal nd-proof $\mathbf{D_I}$ of G from α,β is:

$$\dfrac{\phi_1 \vee \phi_2 \qquad \begin{matrix}[\phi_1]\\ \vdots \\ G\end{matrix} \qquad \begin{matrix}[\phi_2]\\ \vdots \\ G\end{matrix}}{G}$$

This construction is proper, as $\downarrow$**v** has as its major premise an element of α,β and G is the endformula of $\mathbf{D_{J_i}}$.

$\downarrow\rightarrow$: The immediate subtrees $\mathbf{J_1}$ and $\mathbf{J_2}$ of **I** have roots $\alpha;\beta?\phi_1$, respectively $\alpha;\beta,\phi_2?G$; by induction hypothesis there are uniquely determined normal nd-proofs $\mathbf{D_{J_1}}$ and $\mathbf{D_{J_2}}$ leading from α,β to ϕ_1, respectively from α,β,ϕ_2 to G. Use $\mathbf{D_{J_1}}$ and the fact that $\phi_1 \rightarrow \phi_2 \in \alpha,\beta$ to construct a normal proof **D** of ϕ_2 from assumptions in α,β.

$$\frac{\phi_1 \qquad \qquad \phi_1 \rightarrow \phi_2}{\phi_2}$$

If $\mathbf{D}_{\mathbf{J}2}$ contains any occurrences of ϕ_2 as open assumptions, then replace those occurrences by **D**. This construction yields the normal proof $\mathbf{D}_\mathbf{I}$ of G from assumptions in α,β.

Now let me treat the ↑-rules. ↑∧: The immediate subtrees $\mathbf{J}_i$ of **I** have roots $\alpha;\beta?G_i$, for i=1 or 2, and G is $(G_1 \wedge G_2)$; by induction hypothesis there are uniquely determined normal nd-proofs $\mathbf{D}_{\mathbf{J}i}$ leading from α,β to G_i. The nd-proof $\mathbf{D}_\mathbf{I}$ is obtained by joining $\mathbf{D}_{\mathbf{J}1}$ and $\mathbf{D}_{\mathbf{J}2}$ via ∧-introduction.

↑$\mathbf{v}_i$: The immediate subtree $\mathbf{J}_i$ of **I** has root $\alpha;\beta?G_i$ and G is $(G_1 v G_2)$; by induction hypothesis there is a uniquely determined normal nd-proof $\mathbf{D}_{\mathbf{J}i}$ leading from α,β to G_i. The normal nd-proof $\mathbf{D}_\mathbf{I}$ is obtained by v-introduction.

↑→: The immediate subtree **J** of **I** has root $\alpha,G_1;\beta?G_2$ and G is $(G_1 \rightarrow G_2)$; by induction hypothesis there is a uniquely determined normal nd-proof $\mathbf{D}_\mathbf{J}$ leading from α,G_1,β to G_2. The nd-proof $\mathbf{D}_\mathbf{I}$ is obtained by →-introduction with G_1 and G_2.

Finally, I treat the rules for negation. $\bot_i$: The immediate subtree **J** [$\mathbf{J}_\neg$] of **I** has root $\alpha,G_1;\beta?[\neg]\varphi$, where G is $\neg G_1$ and $\varphi \in \mathbf{F}$; by induction hypothesis there are uniquely determined nd-proofs $\mathbf{D}_\mathbf{J}$ and $\mathbf{D}_{\mathbf{J}\neg}$ leading from α,G_1,β to φ, respectively $\neg\varphi$. The nd-proof $\mathbf{D}_\mathbf{I}$ is obtained by applying ¬-introduction to infer G. $\bot_\mathbf{c}$: This final rule is treated in the same way as the previous negation rule. **Q.E.D.**

The nd-proof $\mathbf{D}_\mathbf{I}$ uses exactly the same rules as **I**. The structural similarity between ic-derivations and nd-proofs is even more apparent, when the latter are represented graphically by Fitch-diagrams; consider the examples in Diagram 2 of the Appendix. Then the ic-derivations can actually be viewed as prescriptions for constructing isomorphic Fitch-diagrams. (For the implementation of proof search in the Proof Tutor, Scheines and I chose Fitch-diagrams to represent nd-proofs, not Gentzen-Prawitz-trees; cf. Notes 1 and 8.) Joining the proposition and the earlier observation concerning the extraction of ic-derivations from ic-trees we have:

Proof extraction theorem. For any a and G: If the intercalation tree a for a;?G evaluates as Y, then a normal nd-proof of G from assumptions in a can be found.

It is extremely easy to obtain the interpolation theorem (and other metamathematical results); the argument is a modification of that for the proof extraction theorem.

Interpolation theorem. For any a, G: if G is a logical consequence of a, then there is an interpolating formula F together with normal nd-proofs Dp and Dp,G/ such that Dp leads from a to F, and Dp,G leads from F to G.

The theorem follows from the next proposition, when observing (with the counterexample extraction theorem established in the next section) that — on account of the fact that G is a logical consequence from a — the ic-tree for the question a;?G evaluates as Y and thus contains an ic-derivation answering the question a;?G.

Proposition: For any I, a, P, G: if I is an ic-derivation for a;p?G, then there is a uniquely determined interpolant F, an nd-proof Dp leading from oc,P to F, and an nd-proof Dp,G leading from F to G. Furthermore, Dp and Dp,G are normal.

5. *Extracting Counterexamples.* By the evaluation of ic-trees we know that a question a;?G obtains the value Y or N. In case the value is Y we can determine an associated normal proof. In case the question has value N, we have as an immediate consequence: "The search failed!" But that only means that the particular possibilities of building up derivations — as reflected in the construction of the ic-tree ~ do not lead to a proof establishing G from assumptions in a. We will do better: a *specially* selected branch in the ic-tree can be used to define a semantic counterexample to the inference from a to G. Clearly, if the question cc;?G evaluates to N, so does the question a,G';?l, where G^1 is ^G if G is not a negation and is its unnegated part otherwise. So we establish:

Counterexample extraction theorem. For any a and G: If the intercalation tree a for a,G^f;?l evaluates as N, then it contains a canonical refutation branch p that determines a valuation v with v*=<|> for all <|>€a,G\ (That is, v is a counter-example to the inference from a to G.)

As the ic-tree a evaluates to N it will be quite direct to see that the following construction leads to a branch p through a if F(a,G') is non-empty. If this set is empty, a,G' consists only of sentential letters, and the valuation v, defined by vtP iff P€OC,G\ is a counterexample. If the set of proper sub-formulas of the elements of a,G^f is non-empty, we need a more sophisticated argument and, naturally, some auxiliary definitions. The finite set F(a,G') for the negation rule 1y can be enumerated (without repetition) by <HJ>J€|, where l:= {i | l≤i≤n}; let Ho be G^1. For i€| define first of all

K(P,i) = f \&.. (i≤k≤n A *H*\& P A -,Hk« P) if there is such an H_k
0 otherwise

The sequence of nodes of p (and more) is determined as follows:

p*(0) = <x;?G
oto = a,G'
$\lambda(\alpha_0,1)$ = K(OCO,1)
$\rho^*(1)$ = ao;?l
if 0<m:
p*(2m) = { a_m-i;?Hx(a0,m) if [am-i;?Hx($_a$o,$_m$)] = N
«m-i;?-¬$H_{\lambda(\alpha0,m)}$ otherwise

α_m = { $\alpha_n$vi , ->Hx(aO,m) if goal of p*(2m) is Hx($_a$o,m)
α_{m-1} , $H_{\lambda(\alpha0,m)}$ if goal of p*(2m) is -iHx(aO,m)

X,(ao,m+1) = K(a_m,X(ao,m))
p*(2m+l) = a_m;?l

Let v be the smallest m with X,(ao,m+l)=O; then a_v = a_{v+}i, and the refutation branch p is the restriction of p* to {m I mS2v+l}. Let me illustrate and clarify this construction through Diagram 3 in the Appendix: At each step in selecting the next node of the canonical branch p one or the other indicated possibility of proceeding must obtain (as long as the set of assumptions can be properly extended), because not both conclusions of if with the contradictory pair Hk and -iHk can be evaluated as Y. (In case both are evaluated as N, choose the leftmost.) So we have selected a branch p through the ic-tree a that is N-closed, all of whose nodes evaluate to N, and whose "closing node", indicated by the checkered rectangle, is such that no rule other than if is applicable. Application of that rule with any formula in F(a,G[f]), in particular with Ho, leads to the canonical closing indicated in the diagram.

Let T:= {i} | <f>€a$_v$+i}; thus, r consists of all the formulas appearing on the l.h.s. of the question mark at p[f]s top node. The set F has important syntactic closure properties and this can be exploited to define a valuation that will serve as a model for a, G[1]. We establish first the closure properties.

Closure lemma. For all formulas \|/:

(i) \|/ is a subformula of an element in F => y € r or -i\j/ € r , but not both;

(ii) \jf is —i—i<J>i, —i—i<J>i^r => $\phi_1 \in \Gamma$;

(iii) \|/is (<1>IA<J>2) , (<|>iA<t>2)€r => <j>i[€]F and $\phi_2 \in \Gamma$;
\\f is -I(<|>IA<|>2), -«(<l>iA<|>2)€r => -i<j>i€r or $\neg\phi_2 \in \Gamma$;

(iv) Yis(<t>iV<l>2) , (f a v ^ ^ F => <)>i€r or <J>$_2$€F;
\j/ is —i(<t>iV<|)2) ,-i(<hv<|>2)€r => —i<J>i€r and $\neg\phi_2 \in \Gamma$;

(v) **\|/is((l)i^(()2),((|)i-^(|)2)^r => $\neg\phi_1 \in \Gamma$ or $\phi_2 \in \Gamma$;**
V is-i^i-xfc) ,-i(<t>i->4>2)[€]r => <|>i€r and -i<J>$_2$€F.

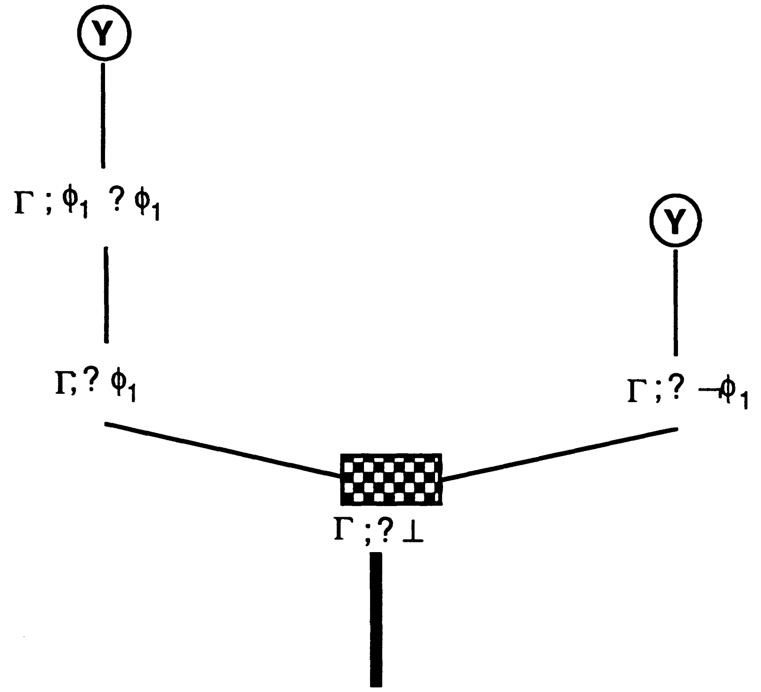
Proof, (i) is direct from the construction of F. (ii) is an almost immediate consequence of (i): Assume —i-i^F and, to obtain a contradiction, <j)i$F; from the second assumption and the first part of (i) it follows that -i<j>i£ F. But that together with the first assumption contradicts the second part of (i). — Now let me establish (iii) to emphasize the pattern of argumentation. We have to show:

(*) (<l>iA<|>2)er=> ^*er* and $\phi_2 \in \Gamma$

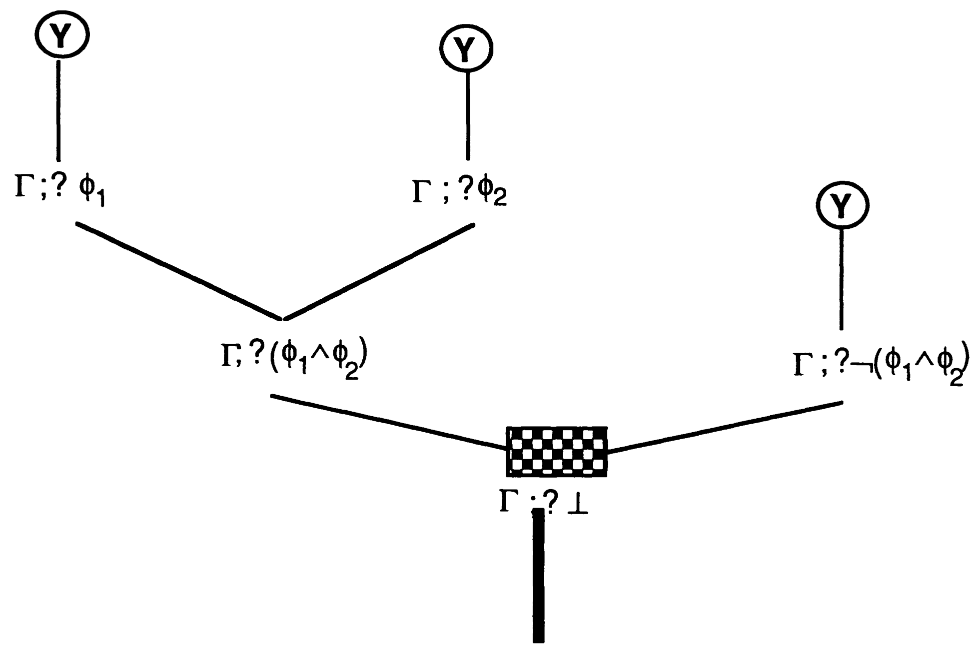
and

(**) -i(4>iA<te)€r =>-.<t>i€F or $\neg\phi_2 \in \Gamma$.

For (*) assume ((^lA^JeF and <(>i$r (the case <t>$_2$$F is symmetric); by (i) $\neg\phi_1 \in \Gamma$. Given these conditions we can close the branch as follows, applying lA_a to the left node above the checkered one :



This contradicts the fact that the checkered node is evaluated as **N**. (**) is established in an analogous way applying $\uparrow\wedge$ instead of $\downarrow\wedge_1$: Assume that $\neg(\phi_1\wedge\phi_2)\in\Gamma$, $\neg\phi_1\notin\Gamma$, and $\neg\phi_2\notin\Gamma$; from the last two assumptions and (i) follows $\phi_1\in\Gamma$ and $\phi_2\in\Gamma$, and the branch can be closed as indicated in the next diagram.



Q.E.D

Now define a valuation by $\mathbf{v}\models P$ *iff* $P\in\Gamma$. Using this valuation and the closure lemma we can prove the **Proposition**: for every $\phi\in\Gamma$, $\mathbf{v}\models\phi$. Hence **v** is a model for α, G'; this concludes the proof of the theorem concerning the extraction of counterexamples. Putting these considerations together, we have a completeness theorem for classical sentential logic in the following form:

Completeness theorem. The ic-tree for the question α;?G allows us to determine either a normal proof of G from α or a branch that provides a counterexample to the inference from α to G.

This yields, as far as I know, the first *semantic proof of the normal form theorem* for a natural deduction calculus.

Normal form theorem. If G can be proved from assumptions in α, then there is a normal proof of G from α.

Remark. For *proof search* it is important that the ic-tree be pruned -- without losing completeness. One way of achieving that is by restricting the formulas with which contradictory pairs can be formed; one can do that in at least three (successively more restrictive) versions using positive, strictly positive, and directly available negations only. The definitions of positive and strictly positive subformulas are given in the Appendix. In the first modification $\mathfrak{P}(\alpha)$ is the class of all **positive** subformulas of elements in α.

$\bot_{\mathfrak{P}}$: $\alpha;\beta?\bot,\ \neg\varphi\in\mathfrak{P}(\alpha)\ \Rightarrow\ \alpha;\beta?\varphi$ AND $\alpha;\beta?\neg\varphi$.

In the second modification $\mathfrak{S}(\alpha)$ is the class of all **strictly positive** subformulas of elements in α.

$\bot_{\mathfrak{S}}$: $\alpha;\beta?\bot,\ \neg\varphi\in\mathfrak{S}(\alpha)\ \Rightarrow\ \alpha;\beta?\varphi$ AND $\alpha;\beta?\neg\varphi$.

In the final formulation a negated formula must be directly available; the bracketed part leads clearly to immediate **Y**-closure.

$\bot_{\mathfrak{D}}$: $\alpha;\beta?\bot,\ \neg\varphi\in\alpha,\beta\ \Rightarrow\ \alpha;\beta?\varphi$ [AND $\alpha;\beta?\neg\varphi$].

To establish completeness for each of the resulting variants of the ic-calculus it suffices to show that the modified ic-trees (built up by the ↑- and ↓-rules, $\bot_c$, $\bot_i$, and one of the modified rules $\bot_{\mathfrak{O}}$) allow the extraction of a counterexample in case [a;?G]=N. The construction of a canonical refutation branch involves now not only the indirect rules, but all the other rules. In defining such a branch one has to make sure that the appropriate version of the closure lemma can be established. For the calculus involving Ip that means, for example, that clause (i) is replaced by (lj>): y is a positive subformula of an element in T => y € r or -i\|f € r, but not both. It should be noted that the rule that prunes the full ic-tree most, i©, is not the most useful for proof search: The other rules direct the search in significant ways.

6. Problem space for predicate logic. The metamathematical considerations for sentential logic can be extended to predicate logic. To that end I use the following formulation of the E- and I-rules for the quantifiers; note that writing <|>t assumes that t is free for x in <|>x or, alternatively, that some bound variables in (|>x have been renamed. For V we have the rules:

VE

(Vx)(])x
<|>t

VI

<|x
(Vx)<|>x

Applications of the I-rule must satisfy the restriction that x does not have a free occurrence in any assumption on which the derivation of <|>x depends. Note that the inference from *tyx* to (Vy)<J>y, for sufficiently new y, is a derived rule. — For 3 we have the rules:

3E

[ϕy]
.
.
(∃x)ϕx r|
n

31

^t
(3x)<|>x

with the usual restriction on the E-rule, namely, y must not have free occurrences in r| or (3x)<|>x nor in any assumption (other than <))y) on which the proof of (the upper occurrence of) T| depends.

To build up ic-trees one applies now also quantifier rules "to close the gap between assumptions and conclusion" in the ic-format. In the formulation of the ic-rules $\mathbf{T}(\gamma,G)$ denotes the finite set of terms occurring in the formulas of γ,G.

$\downarrow\forall$: $\alpha;\beta?G,\ (\forall x)\phi x\in\alpha,\beta,\ t\in\mathbf{T}(\alpha,\beta,G),\ \phi t\notin\alpha\beta \Rightarrow \alpha;\beta,\phi t?G$

$\downarrow\exists$: $\alpha;\beta?G,\ (\exists x)\phi x\in\alpha\beta$, y is new for $\alpha,(\exists x)\phi x,G$, and there is no $t\in\mathbf{T}(\alpha,\beta,G)$ with $\phi t\in\alpha\beta \Rightarrow \alpha,\phi y;\beta?G$

$\uparrow\forall$: $\alpha;\beta?(\forall x)\phi x$, y is new for $\alpha,(\forall x)\phi x \Rightarrow \alpha;\beta?\phi y$

$\uparrow\exists$: $\alpha;\beta?(\exists x)\phi x,\ t\in\mathbf{T}(\alpha,\beta,G) \Rightarrow \alpha;\beta?\phi t$

Ic-trees are specified inductively as in the case of sentential logic: if $\alpha^*;\beta^*?G^*$ is an open question, all possibilities of intercalating formulas are considered. In case G^* is different from $\bot$ (and the question is not a repeat question) one proceeds, e.g., in the order $\downarrow\forall$, $\downarrow\&_1$, $\downarrow\&_2$, $\downarrow\rightarrow$, $\downarrow\exists$, $\downarrow\mathbf{v}$, $\uparrow\forall$, $\uparrow\&$, $\uparrow\rightarrow$, $\uparrow\exists$, $\uparrow\mathbf{v}_1$, $\uparrow\mathbf{v}_2$, and finally either $\bot_i$ or $\bot_c$; in case G^* is $\bot$ we apply $\bot_{\mathcal{F}}$ with $\mathcal{F}$ containing all proper subformulas of α^* (where subformulas of quantified formulas are taken only with terms in $\mathbf{T}(\alpha^*,\beta^*,\bot)$). Branches are closed with **Y** and **N** under the same conditions as before. In general, however, ic-trees will not be finite. Thus, at every stage of construction there may be an open question at some leaf; to evaluate finite *partial* ic-trees σ^* a third value **U** is assigned to such a leaf. Given the valuation $\mathbf{v}_{\sigma^*}$, the value of the question at σ^*'s root is determined by recursion on σ^* following Kleene's scheme [IM, p. 334] for three-valued logic:

$$[N]_{\sigma^*} = \mathbf{v}_{\sigma^*}(N) \qquad \text{if N is a leaf of } \sigma^*$$

in case N is the unique successor of M

$$[N]_{\sigma^*} = [M]_{\sigma^*}$$

in case N is at a conjunctive branching,

$$[N]_{\sigma^*} = \begin{cases} Y & \text{if for all immediate predecessors M of N: } [M]_{\sigma^*}=Y \\ N & \text{if for some immediate predecesor M of N: } [M]_{\sigma^*}=N \\ U & \text{otherwise} \end{cases}$$

in case N is at a disjunctive branching,

$$[N]_{\alpha}^{*} = \begin{cases} N & \text{if for all immediate predecessors M of N: } [M]_{\alpha}^{*}=N \\ Y & \text{if for some immediate predecesor M of N: } [M]_{\alpha}^{*}=Y \\ U & \text{otherwise} \end{cases}$$

The full ic-tree α for α;?G is defined in stages as follows: α_0 is α;?G; α_{n+1} is α_n if $[\alpha;?G]_{\alpha_n}$ is either Y or N, otherwise α_{n+1} is obtained from α_n by expanding each open branch by all applicable rules. Three possibilities can arise: (1) for some n∈N $[\alpha;?G]_{\alpha_n}$=Y, (2) for some n∈N $[\alpha;?G]_{\alpha_n}$=N, and (3) for all n∈N $[\alpha;?G]_{\alpha_n}$=U. In the first case a normal derivation can be associated with a subtree of α_n — by selecting an ic-derivation and by proving (inductively) that each ic-derivation determines a unique normal derivation of G from elements in α. In the second case we can construct a finite canonical refutation branch as in sentential logic and define from it a counterexample. The third case, whose treatment is clearly crucial to complete this sketch of the completeness proof, requires additional considerations.

7. *Completeness and normal form.* The extraction of a counterexample from an infinite ic-tree requires some circumspection: Instead of constructing a refutation branch directly, we determine first a particular infinite subtree of the ic-tree and then apply Koñig's Lemma to this *canonical refutation tree.*

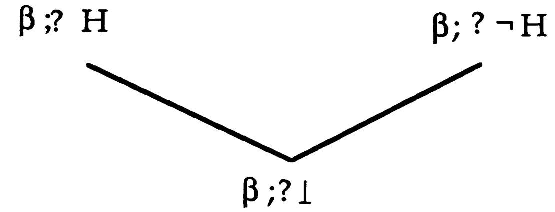
Counterexample extraction. For any α and G: if the intercalation tree α for α;?G is such that for every natural number n $[\alpha;?G]_{\alpha_n}$=U, then α contains an infinite refutation branch ρ that determines a structure *Tt* with H ^ , for all ϕ in α, and M^-»G. Thus, *ft* is a counterexample to the inference from α to G.

The reason for having to cut down the ic-tree α to the canonical refutation tree x is this: Refutation branches have to satisfy suitable closure conditions, and it is trivial to construct infinite branches of α that don't. So we define x in such a way that all of its infinite branches satisfy the closure conditions. The pertinent considerations extend those for sentential logic with variations on familiar Henkin and "fair" tableaux constructions; thus I emphasize only the crucial points.

The construction of x (as a subtree of the intercalation tree α) for the question α;?G proceeds in two waves: The first aims for "sub-maximization" with respect to a given finite set of formulas, whereas the second introduces new subformulas by witnessing -- through instances with new variables -- existential and negated universal formulas that occur on the l.h.s. of ?. We start out the construction of the binary tree τ (using conventions and definitions from the sentential case) with the first wave for the enumeration of the proper subformulas of formulas in α,G' (where immediate subformulas of quantified formulas are taken only with terms in $\mathbf{T}(\alpha, G', \perp)$:

$$\tau(0) = \alpha;?G$$
$$\alpha_0 = \alpha, G'$$
$$\lambda(\alpha_0, 1) = \kappa(\alpha_0, 1)$$
$$\tau(1) = \alpha_0;?\perp$$

Now let 0<m; at level 2m we extend each open branch (i.e. its leaf evaluates as **U**) with a question of the form $\beta;?\perp$ at its leaf by



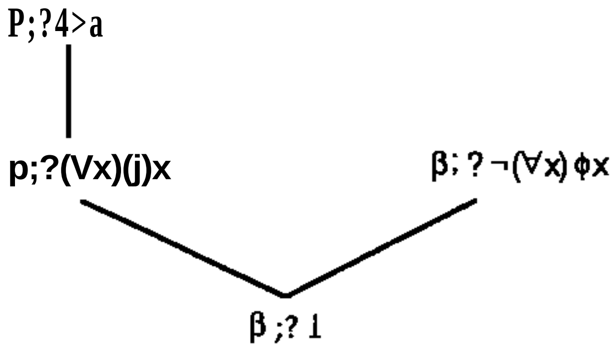
if both questions $\beta;?H$ and $\beta;?\neg H$ evaluate as **U**; if only one of them evaluates as **U**, then the branch is extended at just that question. One of these cases must hold, because the question $\beta;?\perp$ has value **U**. (Clearly, as before, H is the first element in the given enumeration that extends β properly.) At the next level 2m+1, every open branch is extended by applying the appropriate negation rule. After finitely many steps this construction cannot be continued. However, at least one branch in the tree constructed so far has to be open for extensions by rules other than $\perp_{\mathcal{F}}$, as for all $n \in N$ $[\alpha;?G]_{\sigma n} = \mathbf{U}$. In sentential logic, we saw, that cannot happen; the resulting set of formulas Γ is deductively closed in the sense of the earlier Closure Lemma. Here, some of the Γ's associated with leaves cannot satisfy the closure conditions

$(\exists x)\phi x \in \Gamma \Rightarrow \phi t \in \Gamma$ for some term t

and

$\neg(\forall x)\phi x \in \Gamma \Rightarrow \neg\phi t \in \Gamma$ for some term t.

In the first case the rule $\downarrow\exists$ is applicable (with a canonically chosen new variable); in the second case we are able to extend the branch in the following way (and also with a canonically chosen new variable):



The right extension closes with Y, whereas the left one remains open. This brings us to the second wave: We apply 13 in all needed cases and then perform the above analysis on those -«(Vx)<|>x for which no negated instance is available. The first wave can be repeated now for an extended set of formulas and so on, obviously! We obtain in this way an infinite, binary subtree x of the ic-tree; König's Lemma applied to x yields an infinite branch p. Define F_p = (V I V occurs on the l.h.s. of ? in some question on p}; this set has all the appropriate closure properties needed to serve as the basis for the counterexample definition. Let $T(r_p)$ consist of all terms that occur in some formula of F_p. (Subformulas are defined with respect to this set of terms.)

Closure lemma. For all formulas \|/:

(i) \j/ is a subformula of an element in F_p => V^F_p or -lyeFp , but not both;

(ii) vis-i-i<|>i,-i-i<|>i€rp => $\phi_1 \in \Gamma_\rho$;

(iii) \|/is(<hA(f>2) ,(<l>iA(|>2)€r_p => 4>i€r_p and $\phi_2 \in \Gamma_\rho$;
V is -(<l>iA<t>2), -i(<|>iA<k)€r_p => -i<h€r_p or $\neg\phi_2 \in \Gamma_\rho$;

(iv) v is (4>iv4>2), (4>iv4^2)€r_p => <|>i€$r_p$or $\phi_2 \in \Gamma_\rho$;
y is-i(<t>iv<t>2) ,-i(^iv^2)€r_p => $\neg\phi_1 \in \Gamma_\rho$ and $\neg\phi_2 \in \Gamma_\rho$;

(v) \|^is (<t>i-^<t>2), (<t>i-^ct>2)^rp => $\neg\phi_1 \in \Gamma_\rho$ or $\phi_2 \in \Gamma_\rho$;
$\phi_1 \in \Gamma_\rho$ and $\neg\phi_2 \in \Gamma_\rho$.

(vi) \|f is (3x)<|>x , (3x)(|)x€r_p => 4>t€r_p for some term t€$T(r_p)$;
\j/is-,(3x)(|)x,-n(3x)(l)x€r_p => -i(()t€r_p for all terms t€$T(r_p)$;

(vii) \\f is (Vx)<|>x , (Vx)(^x€r_p => (()t€r_p for all terms t€$T(r_p)$;
\|/is-.(Vx)(t>x,-i(Vx)(|)x€r_p => -i<(>t€r_p for some term t€$T(r_p)$.

The definition of a structure *Tt* from F_p is now standard, and we obtain a completeness theorem for classical predicate logic in the form:

Completeness theorem. The ic-tree for the question a;?G allows us to determine either a normal nd-proof of G from a or a branch that provides a counter-example *Tt* to the inference from a to G.

So we have a semantic argument for the normalizability of nd-proofs; and from ic-derivations we can construct not only nd-proofs, but also as in the case of sentential logic interpolants to obtain the interpolation theorem.

Normal form theorem. If G can be proved from assumptions in a, then there is a normal nd-proof of G from a.

Remarks (continuing remark in section 5)[8]. (1) The indirect rule ly can be restricted as before: to positive, strictly positive, and directly available formulas. For proof search it is best to balance width and depth of vision, most likely accomplished (in the case of classical logic) by focusing on negated positive subformulas. (2) For the search algorithm the language of predicate logic is expanded by "new free variables" and Skolem (and Herbrand) functions as done, for example, in Fitting's book. It is in this expansion that quantifiers are eliminated during the search in a "canonical" way. To direct the search we use heuristics employed for sentential logic together with two novel features, namely an appropriately narrow concept of "positive canonical subformula" and a unification algorithm for quantified formulas, see [Sieg and Kauffmann].

8. So what? This work is to address, ultimately, the question of finding proofs in mathematics with logical *and* mathematical understanding. If one looks at Georg Polya's writings on mathematical reasoning and heuristics one realizes quickly that his most general strategies for argumentation are simple logical ones. Clearly, logical formality per se does not facilitate the finding of

---

[8] Quite sophisticated strategies are involved in the algorithm underlying the Carnegie Mellon Proof Tutor that searches automatically for nd-proofs in sentential logic; that program was developed by Richard Scheines and myself with assistance from Jonathan Pressler and Chris Walton. Presently we are redesigning it in collaboration with John Byrnes, and we have extended the search algorithm to predicate logic along the lines sketched here. For details, in particular concerning heuristics, see [Sieg and Scheines].

proofs. Logic within a natural deduction framework does help, however, to bridge the gap between assumptions and conclusions by suggesting *very rough* structures for arguments, i.e. *logical structures* that depend solely on the syntactic form of assumptions and conclusions. This role of logic, though modest, is the starting-point for moving up to subject-specific considerations that support a theorem.

Proofs provide explanations of what they prove by putting their conclusions in a context that shows them to be correct. The deductive organization of parts of mathematics is *the* classical methodology for specifying such contexts. This methodology has two well-known aspects: the formulation of principles, i.e. axioms, and the reasoning from such principles; the latter is mediated through logical inferences and subject-specific lemmata. Heuristic considerations and "leading mathematical ideas" for particular parts of mathematics have to be found and properly articulated. Saunders MacLane (1934) suggested to include in the scope of logic such a *structure-theory of proofs*: this extension of the traditional role of logic and, in particular, of proof theory interacts directly and, I am convinced, fruitfully with a sophisticated, automated search for humanly intelligible proofs.

APPENDIX

In this appendix I give first a definition used in the **Remark** at the end of section 5; then three diagrams are drawn that complement the text of sections 3 and 5.

Positive and *strictly positive subformulas* of a given formula are defined by induction; indeed, for the first concept one defines simultaneously, when ϕ is a *positive subformula* of ψ [$\phi \in \mathbf{P}(\psi)$] or ϕ is a *negative subformula* of ψ [$\phi \in \mathbf{N}(\psi)$], namely by the rules,

(i) ϕ is ψ => $\phi \in \mathbf{P}(\psi)$,

(ii) (a) ψ is $\neg\psi_1$, $\phi \in \mathbf{N}(\psi_1)$ => $\phi \in \mathbf{P}(\psi)$,

(b) ψ is $\neg\psi_1$, $\phi \in \mathbf{P}(\psi_1)$ => $\phi \in \mathbf{N}(\psi)$,

(iii) (a) ψ is $\psi_1 \wedge \psi_2$, $\phi \in \mathbf{P}(\psi_1) \cup \mathbf{P}(\psi_2)$ => $\phi \in \mathbf{P}(\psi)$,

(b) ψ is $\psi_1 \wedge \psi_2$, $\phi \in \mathbf{N}(\psi_1) \cup \mathbf{N}(\psi_2)$ => $\phi \in \mathbf{N}(\psi)$,

(iv) (a) ψ is $\psi_1 \vee \psi_2$, $\phi \in \mathbf{P}(\psi_1) \cup \mathbf{P}(\psi_2)$ => $\phi \in \mathbf{P}(\psi)$,

(b) ψ is $\psi_1 \vee \psi_2$, $\phi \in \mathbf{N}(\psi_1) \cup \mathbf{N}(\psi_2)$ => $\phi \in \mathbf{N}(\psi)$,

(v) (a) ψ is $\psi_1 \rightarrow \psi_2$, $\phi \in \mathbf{P}(\psi_2)$ => $\phi \in \mathbf{P}(\psi)$,

(b) ψ is $\psi_1 \rightarrow \psi_2$, $\phi \in \mathbf{N}(\psi_1)$ => $\phi \in \mathbf{P}(\psi)$,

(c) ψ is $\psi_1 \rightarrow \psi_2$, $\phi \in \mathbf{P}(\psi_1) \cup \mathbf{N}(\psi_2)$ => $\phi \in \mathbf{N}(\psi)$.

Finally, ϕ is a *strictly positive subformula* of ψ [$\phi \in \mathbf{S}(\psi)$] if and only if it can be obtained by just the rules (i), (iii)(a), (iv)(a), and (v)(a).

Diagram 1 illustrates the construction of an ic-tree in an interesting case, namely the proof of tertium non datur. Diagram 2 contains Fitch diagrams for the three proofs of tertium non datur discussed in section 3. Prawitz (1965, 98-99) asserts that already Jaskowski introduced this representation in the late twenties. In any event, for computer implementation Fitch diagrams are convenient for the representation of nd-proofs: they reflect dependencies as graphically as trees do, but are easier to put on a screen and avoid the duplication of parts of proofs necessary in tree representations. Finally, Diagram 3 illustrates the construction of canonical refutation branches discussed in section 5.

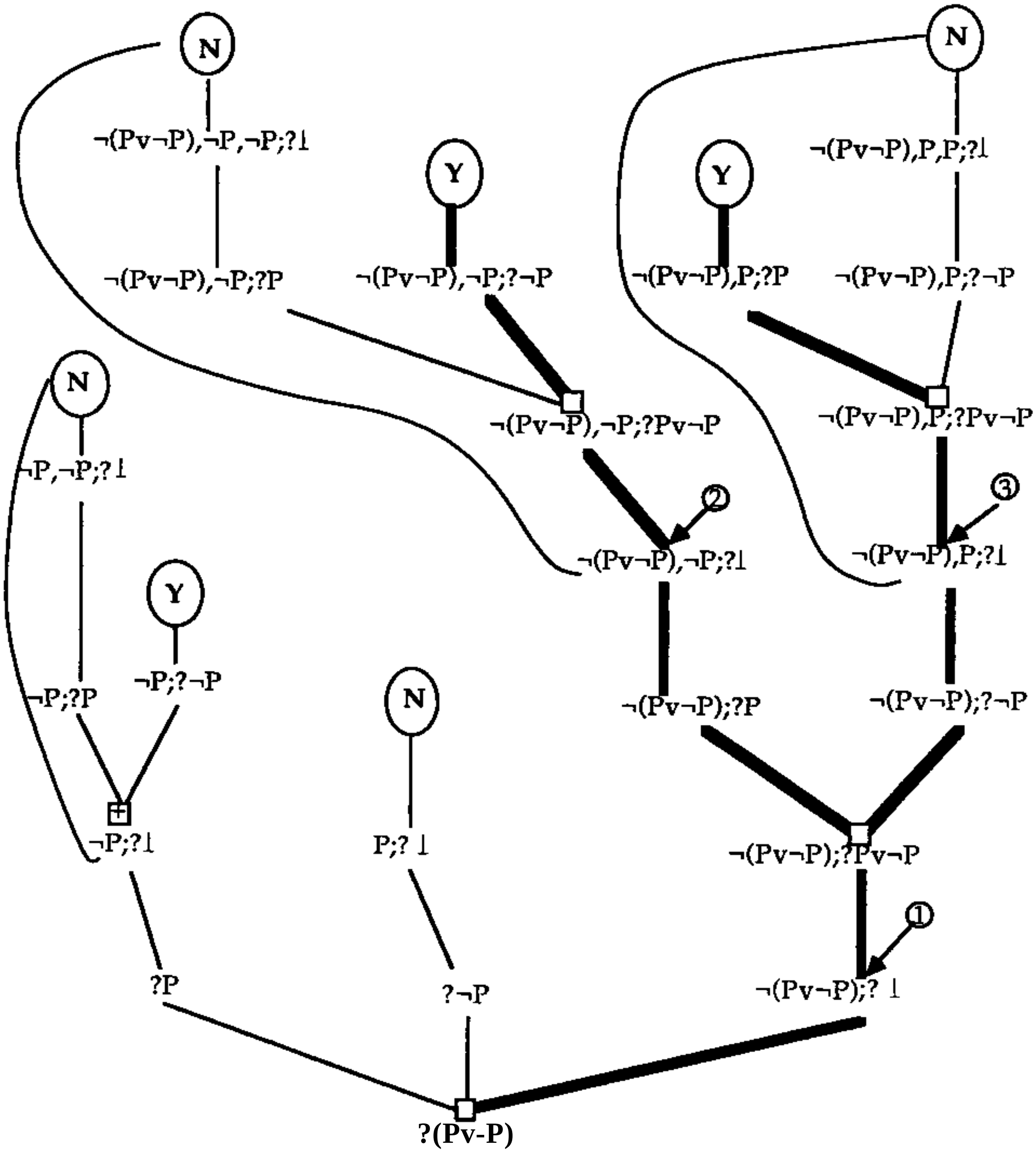

N
¬(Pv¬P),¬P,¬P;?⊥
¬(Pv¬P),¬P;?P
Y
¬(Pv¬P),¬P;?¬P
¬(Pv¬P),¬P;?Pv¬P
②
¬(Pv¬P),¬P;?⊥
¬(Pv¬P);?P
N
¬(Pv¬P),P,P;?⊥
Y
¬(Pv¬P),P;?P
¬(Pv¬P),P;?¬P
¬(Pv¬P),P;?Pv¬P
③
¬(Pv¬P),P;?⊥
¬(Pv¬P);?¬P
¬(Pv¬P);?Pv¬P
①
¬(Pv¬P);? ⊥
N
¬P,¬P;?⊥
Y
¬P;?P
¬P;?¬P
¬P;?⊥
N
P;? ⊥
?P
?¬P
?(Pv-P)

Diagram 1

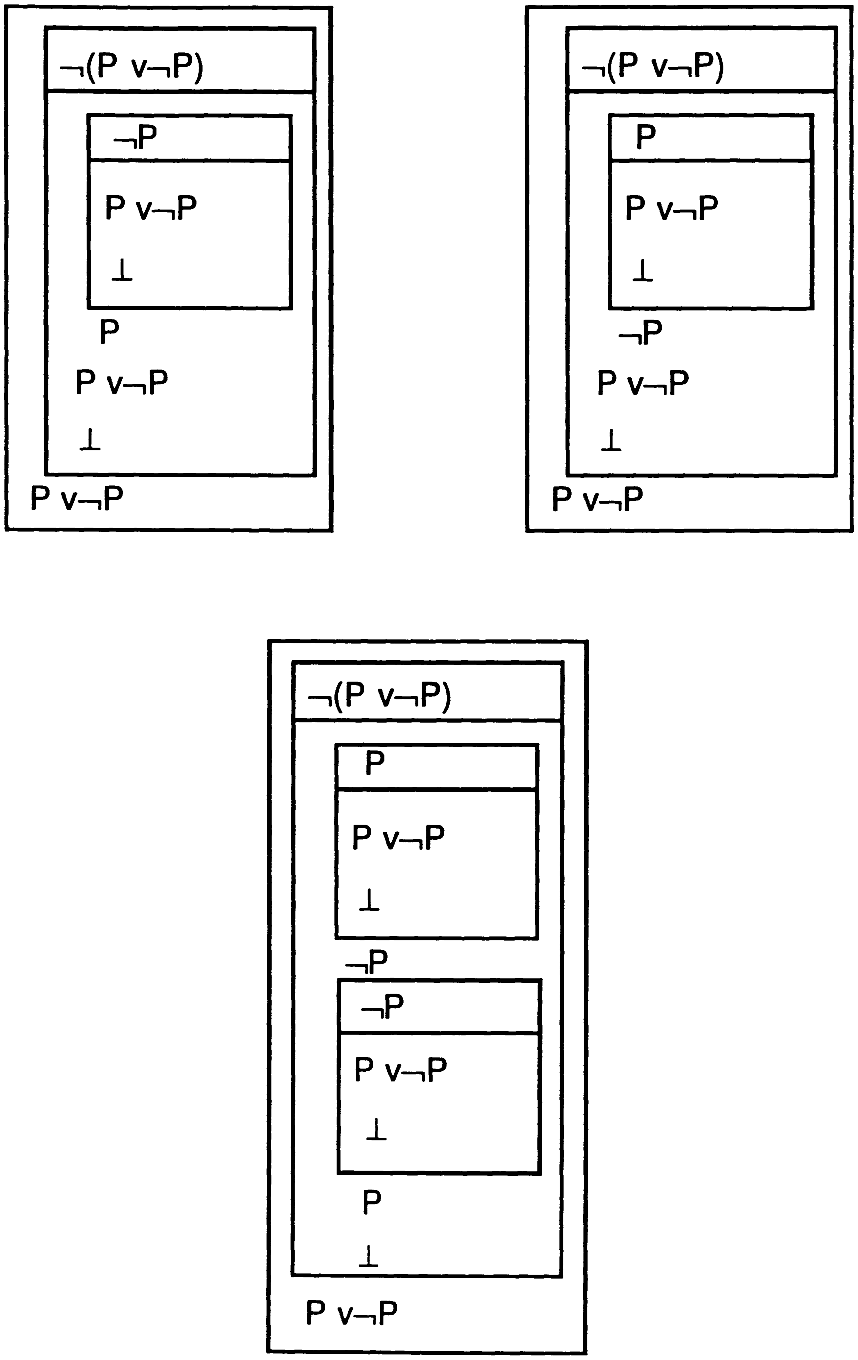


Diagram 2

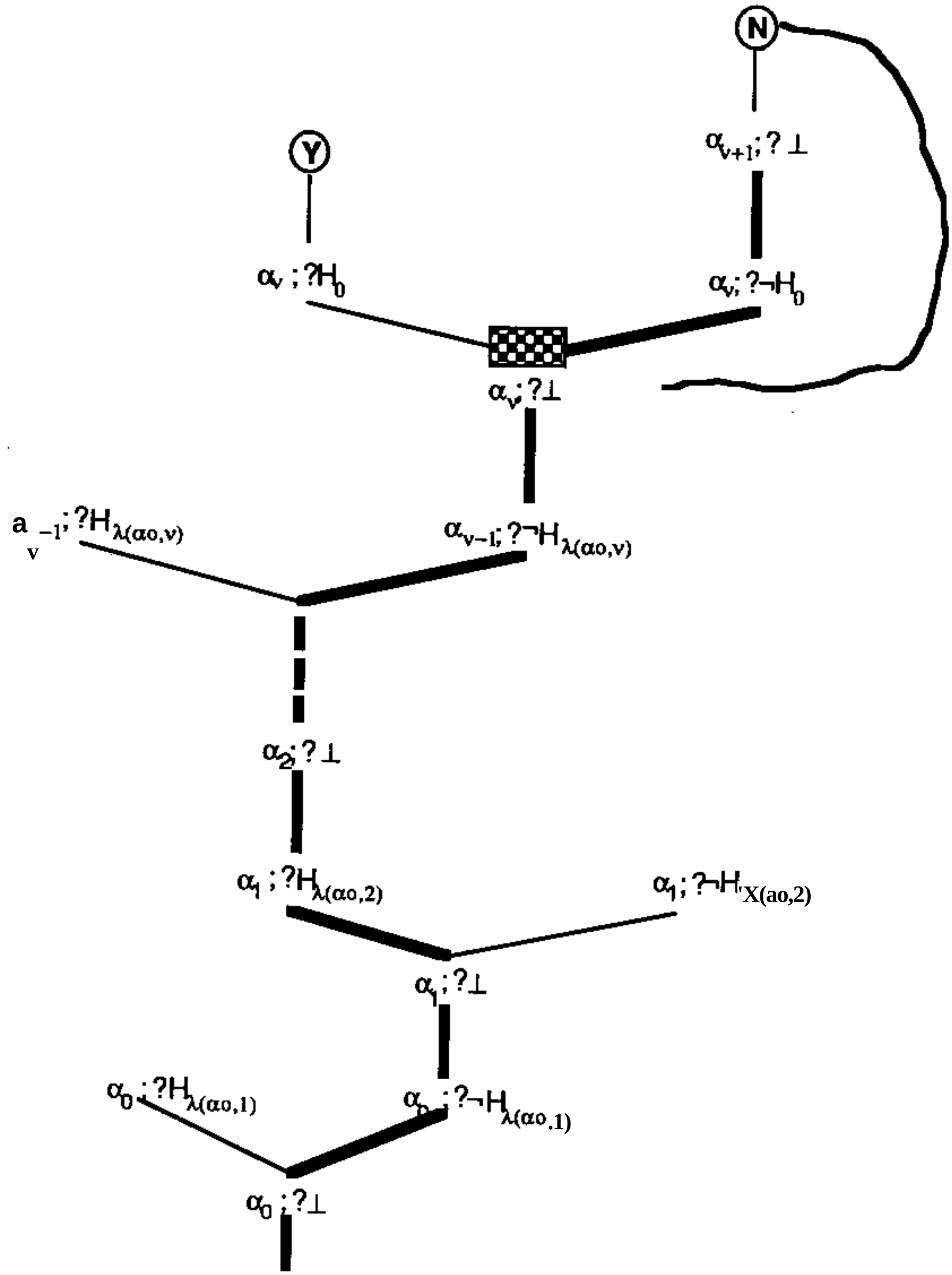

Y
N
αν+1; ? ⊥
αν ; ?H0
αν; ?¬H0
αν; ?⊥
aν−1; ?Hλ(αo,ν)
αν−1; ?¬Hλ(αo,ν)
α2; ? ⊥
α1 ; ?Hλ(αo,2)
α1; ?¬HX(ao,2)
α1; ?⊥
α0 ; ?Hλ(αo,1)
α0 ; ?¬Hλ(αo,1)
α0 ; ?⊥

Diagram 3

References

Andrews, P. "Transforming matings into natural deduction proofs"; 5-th Conference on Automated Deduction"; New York, Berlin: Springer-Verlag 1980, 281-292.

Bledsoe, W. "Non-resolution theorem proving"; Artificial Intelligence 9 (1977), 1-35.
-------------- "The UT natural-deduction prover"; Technical Report, Departments of Mathematics and Computer Science, University of Texas, April 1983.

Cittadini, S. "Intercalation calculus for intuitionistic propositional logic"; Carnegie Mellon Technical Report PHIL-29, Philosophy, Methodology, and Logic; 1992, 34pp.

Fitting, M. First-order logic and automated theorem proving; New York, Berlin: Springer-Verlag, 1990.

Gentzen, G. "Untersuchungen über das Logische Schließen I, II"; Math. Zeitschrift 39 (1934), 176-210 and (1935), 405-431.

Herbrand, J. Logical Writings, W. Godfarb (ed.); Cambridge: Harvard University, 1971.

Kleene, S.C. Introduction to Metamathematics; Groningen: Wolters-Noordhoff Publishing, 1952.

MacLane, S. "Abgekürzte Beweise im Logikkalkül"; Ph. D. thesis, University of Göttingen, 1934.
---------------- "A logical analysis of mathematical structure"; The Monist (1935), 118-130. (The paper was read to the American Mathematical Society on December 28, 1933.)

Nevins, A.J. "A human oriented logic for automatic theorem proving"; J. ACM 21 (1974), 606-621.

Pfenning, F. "Proof transformations in higher-order logic"; Ph.D. thesis, Carnegie Mellon University, 1987.

Prawitz, D. Natural Deduction - proof-theoretical study; Stockholm: Almsqvist & Wiksell, 1965.

Shanin, N.A. e.a. "An algorithm for a machine search of a natural logical deduction in a propositional calculus"; Izdat. "Nauka", Moscow, 1965. Reprinted in: Siekmann and Wrightson (eds.), Automation of Reasoning, volume 1, New York, Berlin: Springer-Verlag, 1983, 424-483.

Sieg, W. and Kauffmann, B. "Unification for quantified formulae"; Carnegie Mellon Technical Report PHIL-44, Philosophy, Methodology, and Logic; 1993, 11pp.

Sieg, W. and Scheines, R. "Searching for proofs (in sentential logic)"; in Philosophy and the computer, L. Burkholder (ed.); Boulder, San Francisco, Oxford: Westview Press, 1992,137-159.

Stalmark, G. "Normalization theorems for full first order classical natural deduction"; J. Symbolic Logic 56 (1991), 129-149.

Szabo, M.E. The collected papers of Gerhard Gentzen; Amsterdam: North-Holland Publishing Company, 1969.