Normal Natural Deduction Proofs (in classical logic) [1]

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

---

[1] The work reported here continues the metamathematical investigations basic for the Carnegie Mellon Proof Tutor, see [Sieg and Scheines]; sections 1–4 are corrected and much improved versions of [Sieg 1992] and [Sieg 1994], whereas sections 5 and 6 expand that work and provide the theoretical basis for automated proof search in predicate logic.

We are grateful for the continued, lively discussions in the weekly meetings of the "CPT-group", i.e., with J. Hughes, M. Ravaglia, and R. Scheines. We also appreciate the comments of R. Dyckhoff, A. M. Conti, and two anonymous referees which helped us to improve an earlier version of this paper.

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 we 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 motivates the formulation of *intercalation calculi.* Ic-calculi are the technical underpinnings for (1) and (2), and our paper focuses on their detailed presentation and meta-mathematical investigation in the case of classical predicate logic. As a central theme emerges the connection between restricted forms of nd-proofe and (strategies for) proof search: normal forms are not obtained by removing local "detours", but rather by constructing proofs that directly reflect proof-strategic considerations. That theme warrants further investigation.

1. Proof Search. Natural deduction calculi have been available since the mid-thirties and reflect ^{tt}as accurately as possible the actual logical reasoning involved in mathematical proofs".[2] They capture the logical structure of arguments, in part, by incorporating *inferences from* and *to* complex formulas with characteristic principal connectives. The rules for the "proper" logical connectives, A, V, —>, V, and 3 are consequently divided into "Elimination", i.e., *proper* E-, and "Introduction", i.e., *proper* I-rules. Rules for negation do not fit fully into this schematic approach, in particular not, if they are formulated in the standard (Gentzen-Prawitz) mould. We use instead a very symmetric formulation: the first rule for negation, $\_L_G$, is the distinctive rule of classical logic and is needed, for example, to prove the law of excluded middle and Peirce's law;

$$\frac{\begin{array}{cc}[\neg\psi] & [\neg\psi]\\ \vdots & \vdots\\ <p & \neg\varphi\end{array}}{\psi}$$

the second rule, -U, captures the form of indirect argumentation admitted also intuitionistically and used, most classically, in the Pythagorean proof of

---

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

the irrationality of $\sqrt{2}$.

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

We consider the rules for negation as both E- and I-rules, but not as *proper* E- or I-rules.

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. Technically, the strategies exploit that the structure of nd-proofs can be made to depend 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 the *major premise*; a proof is called *p-normal*[3], when no formula occurrence in the proof is the conclusion of a proper I-rule or $\bot_c$ and also the major premise of a proper E-rule. To be quite accurate, we have to exclude *segments* of formula occurrences, such that the first formula in the segment is the conclusion of a proper I-rule or $\bot_c$ and the last formula the major premise of a proper E-rule. Here and below we make use of terminology used by Prawitz—with just one exception, we use 'branch' for his 'thread'. Note also that we have not yet defined 'normal'. In order to obtain a definition matching that of Prawitz, we first define the *adjacency condition*: the major premise of a $\bot$-rule must not be inferred by a $\bot$-rule. A *normal* proof, then, is p-normal and satisfies the adjacency condition.

The first central property, the Normalization Theorem, was established by Prawitz for a restricted language[4]: (by a sequence of special "reductions") any proof of G from α in the nd-calculus can be transformed into a normal proof leading from α to G, where α is a sequence of formulas[5]. The sec-

---

[3] This term is not related to the term "p-normal" used by Troelstra and van Dalen.

[4] without $\exists$ and $\vee$; $\bot_c$ was applicable only to get atomic conclusions.

[5] Prawitz's proof for the intuitionistic calculus can be extended to the full classical case with the negation rules formulated in the symmetric way as above; that was established by Byrnes. The strong normalization theorem for the full language (with restricted $\vee$- and $\exists$-inferences) was proved by Statman [1974].

ond central property for the nd-calculus concerns the logical complexity of formulas in proofs: normal proofs E leading from a to G have a (modified) *subformula property,* i.e., every formula occurring in E is (the negation of) either a subformula of G or of an element in a. This is a consequence of the third central property, a structural feature of paths in (the tree presentation of) normal nd-proofs: every path contains a uniquely determined E-part and I-part, consisting only of segments that are major premises of proper E-rules, respectively premises of proper I-rules; these two parts are separated by the minimum segment that is the premise of an I-rule.

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, tableaux,* and *logic programming.* Prom a purely logical point of view this is prima facie peculiar: it is after all the subformula property of special kinds of derivations[6] that makes resolution and related techniques possible; normal derivations in natural deduction calculi, as we just noticed, have that very property with the minor addition mentioned. *Why is it then that nd-calculi have not been exploited for automated proof search?* The answer to this general question seems, in part, to lie in answers to three crucial questions: (1) How can one specify through a calculus *only* 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?

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 the theoretical background for 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, not Gentzen's cut-elimination procedure. Indeed, algorithms for finding *cut-free* derivations are refinements of strategies used in that completeness proof. Such strategies realize the heuristic idea of *searching for semantic counterexamples* and yield trees E such that *either* one of E's branches allows the definition of a coun-

---

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

terexample to "a has *G* as a logical consequence" *or* £ constitutes a cut-free derivation of the sequent -»a, G.[7] 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).

The broad problem is this: How can we derive a conclusion or goal *G* from assumptions ^i,..., *(f>n?* or, more vividly: How can we close the gap between *G* and the <£i,..., <£$_n$ via logical rules? This question is at the heart of spanning search spaces via ic-calculi: their basic rules 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. The ic-calculi provide the underpinning for specifying informal approaches to proof search: their rules are used to construct a search space that contains all possible ways of closing the gap between assumptions and *G* via the ic-rules. In this space we search for a gap-closing subspace that determines, in turn, a unique normal or p-nonnal nd-proof from the assumptions to *G*. If the search fails, the search space contains enough information to yield a semantic counterexample. This sketch of the completeness proof for ic-calculi 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*[8]*; in the case of ic-calculi, one tries to find a proof and, if that search fails, one has a counterexample.* Let us turn to the rigorous metamathematical discussion.

We will discuss at first only classical sentential logic with the connectives -», A, V, →] however, the considerations will then be extended to predicate logic and can be used to treat non-classical logics, for example, intuitionistic

---

[7] -»a consists of the negations of the formulas in a.

[8] 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. Cf. our remark at the end of section 3 and also note 19.

logic.[9] The ic-rules operate on triples of the form *a; plG. 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[10]. To facilitate the description of rules and parts of search trees let us agree on some conventions. Lowercase Greek letters a, /?, 7,... range over finite sequences of formulas; as syntactic variables over formulas we use ^, V> X> • • •> ^{an}d also *G* and *H*; II, E, T,... range over trees. ^ e a expresses that 0 is an element of the sequence a; a, *f3* or a/? is short for the concatenation *a • (3* of the sequences *a* and /?; a, <*f*> stands for the sequence *a * (<f>),* where (0) is the sequence with ^ as its only element. There are three kinds of ic-rules: those corresponding to the proper E-rules for A, V, —>; those corresponding to the proper I-rules for A, V, —>; finally, the rules for negation. Let us list the rules of the first kind, i.e., 4-rules.

A» *I:* a; /??G, <£i A <fe e *a/3* => a; /?, <fc?<3 for i=l or 2

V 4.: a; *p?G,* ^ V ^ e a / J ^ a , <*f*>$_x$\ *p?G* AND a, $\phi_2; \beta ? G$

- 4 : a;0?G, ^i->^Gai8=^a;*F*><*j*>*i* AND a; $\beta, \phi_2 ? G$

Now we formulate the rules that correspond to inverted proper I-rules, i.e., t -rules.

A t: «; £?<£i A <fe => a; /??^i AND a; $\beta ? \phi_2$

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

$\to\uparrow: \alpha; \beta ? \phi_1 \to \phi_2 \Longrightarrow \alpha, \phi_1; \beta ? \phi_2$

Finally, we come to the rules for negation:

$\bot_c(\mathcal{F})$: «;0?G,¥> € ,F(a,-#) =^ a,-<?;^v? AND a, $\neg G; \beta ? \neg\varphi$

$\bot_i(\mathcal{F}$\. *a; 0?^G, (p e T(a, G)* => a, G; *P?<p* AND a, G; *p?-«p*

---

[9] 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, "Normal Natural Deduction Proofs (in non-classical logics)".

"The reason for this separation is that some important syntactic constructions will refer only to the available assumptions; for example, concerning the indirect rules and, later on in predicate logic, concerning the analogue of V-introduction.

$\mathcal{F}(\gamma)$ is obtained as follows. Let F_1 consist of all proper subformulas of formulas in 7 and of all negations occurring in 7. ^(7) then consists of all unnegated formulas in F_1 and the unnegated part *rp* of all negations -»^ in F_7. *T(j)* is obviously finite; that is crucial for the finiteness of the search space. Operations *O* leading to smaller and yet sufficient classes can be specified; cf. the end of section 3. The different calculi we are considering are distinguished through the operation *O,* and we denote a particular calculus by $IC_0(O)$, or simply *IC(O)*—as long as it is clear that we are dealing with sentential logic; the corresponding systems for first order logic will later be denoted by ICi(O).

Remarks. (1) Intuitionistic versions of ic-calculi are obtained by using the rule *ex falso quodlibet a;p?*G_y *<p* € O(a) =* *a;/3?<p* and a; £?-»<? instead of $\pm_C(O)$. For the classical system IC(,F), the rule -4 can be weakened to a; */3?G,* ^ 4 (^ 6 a/?, $<f>_x$ 6 *a/3* => a; /?, ^2?G. But this formulation, as Cittadini noticed, is too weak for intuitionistic logic (and unnatural for proof search even in the classical case).

(2) We formulated the ic-rules as Post-productions, but they can also be represented in the standard way with appropriate side conditions; however, the natural application of these rules is "bottom-up". Here are three reformulations: ^

$$\rightarrow\downarrow: \quad \frac{\alpha;\beta?\phi_1 \qquad \text{ł},\phi_2?G}{a\$G} \qquad \text{with } * \rightarrow \phi_2 \in \alpha\beta$$

$$\rightarrow\uparrow: \quad \frac{}{a;\beta?\phi_1 \rightarrow \phi_2}$$

•U(*): $\overline{\ \text{"}\ }$ ^ ^ g $\overline{\ \wedge\ }$ just i n case $\varphi \in \mathcal{F}(\alpha, G)$

Because of this correspondence we call the consequent(s) of a Post-production, *premise(s)* of the appropriate rule. This reformulation brings out the restrictive character of the 4r-rales: the principal formula of a ^-inference must already be in *a(3.*

Next we turn to the construction of the search or problem space, using these rules; indeed, we shall interleaf the nodes of a tree-like arrangement of questions with "rule nodes" that provide information on the rule that is connecting the questions.

2. The Problem Space for Sentential Logic. As an example of how the ic-rules are used to build up the search space for a question a?G, let us show the search tree for the question *?PV->P.* It is partially presented in Diagrams 1, l.A, and l.B of the Appendix. We start out by applying the three possible ic-rules to obtain new questions, namely, ?P or ?-\P or, proceeding indirectly, "•(PV-.P)?v? and -«(PV-nP)?-.<p with each element *(p* of ,F(-«(PV-«P)). Let us pursue the leftmost branch in the tree. To answer ?P we have to use $\pm_c$ and, because of the restriction on the choice of contradictory pairs, we have only to ask -iP?P and -iP?-«P. In the first case only $\pm_c$ could be applied, but would lead to the question we just analyzed. Thus we *close this branch* with N. In the second case the gap between assumption and goal is obviously closed, so we top this branch with Y. No rule is applicable to the question ?-«P; so that branch is closed with N as well. The other parts of the tree are constructed in a similar manner. Each application of J_{c} (J_i) is labeled "$\pm_c$, <£" ("-Li, *4>")*, where *<f>* is the minor premise of the rule application. The subtree in diagram l.A is not full, but at the numbered nodes 1 through 4 the resulting trees do not help in closing the gap. In contrast, the subtree in diagram 1.B is of interest, and we discuss it below.

The composition of Diagrams 1, l.A, and l.B contains enough information for the extraction of derivations in a variety of styles of natural deduction. For our calculus%e can easily obtain corresponding derivations; namely, first:

$$\frac{\dfrac{[\neg P]}{P \vee \neg P} \quad [\neg(P \vee \neg P)]}{P}$$

$$\frac{\overline{\text{PV-.P}} \qquad [\overline{\text{-.(P V ->}P)}]}{PV{\wedge}P}$$

(Here we use square brackets to indicate cancelation of an assumption.) The second derivation is "dual" to this one with the roles of P and *->P* interchanged. Finally, the derivation that emerges from Diagram l.B:

$$\frac{\dfrac{\dfrac{[\neg P]}{Pv\text{-},p} \quad [\text{-.(P v ->}p))}{P} \qquad \dfrac{\dfrac{[P]}{P \vee \neg P} \quad [\neg(P \vee \neg P)]}{\neg P}}{P \vee \neg P}$$

The proof represented in the second diagram above is p-normal, but it is *not* a normal proof, as the major premise *->P* of the last inference with rule $\underline{I}_C$

has been obtained by J_|. (Natural normalization steps reduce this derivation either to the first derivation or its dual.)

The full search or ic-tree is specified inductively by applying ic-rules to the initial question or to the "non-terminal" leaves of an already obtained partial search tree—in all possible ways, unless the application of a rule leads to a question that is not *new for* the branch determined by the appropriate leaf (a; *(3?G* is the *same question as* a*; *0*?G* just in case the sets of formulas in the sequences *a/3* and *a*fi are identical.) In either case one addresses questions of the form a; */3?G* at a particular node:**

if *G* is an element of a/?, then close the branch determined by the current question node with Y;

if *G* is not an element of *ctfi* and every applicable rule leads to a question that is not new for the branch determined by the current question node then close with N;

if *G* is not an element of *a/3* and some applicable rule leads to a new question, then extend the tree at the current question node for all such rules by appropriate rule and question nodes (with a fixed ordering of rules)[11].

For any implementation of a proof search procedure it is crucial to decide quickly, whether^ particular rule will lead, at the current question node, to a new question or not. A first easy step is to impose local side conditions on the 4-rules that prevent the application of a rule, in case it does lead to the same question; this can be done, for example, as follows:

Ail: a; p?G,<f>i A<f>2 e a/3,<f>i <? a/3 => a; ptfalG for t = 1 or 2

V 4;: a;^?G,^ V *fa e afrfa t afafa <£ a/3 =>* a,<h;/??G AND $\alpha, \phi_2; \beta?G$

- 4: or, j8?G,*i -> <*h* € aftfc * *afafa^G => a;p?</>*$_x$ AND a; $\beta, \phi_2?G$

Indeed, these local side conditions are now taken as part of the ic-rules. A second, more intricate step involves a careful analysis of the conditions under which repeated questions can occur. This allows us to avoid checking for repetitions in many instances. A third step would restrict the application of the indirect rules: JL$_C$ is never applied to negated formulas. Thus, to a given

---

[11] **For example we could use the order A$_x$ i, A$_2$ 4, ->4, V |, A t, ->t* Vi t, V$_2$ t, -U> J_c. We also need to order multiple applications of each rule, say by the order in which** the formulas to which it is applied appear in *a/3*.

question node only one l_-rule is applied. We do not pursue such issues in any systematic way, as we are intending to present only the broad theoretical framework for proof search via ic-calculi; there will be some additional remarks at the end of sections 3 and 4.

The ic-tree is constructed in the above general way for questions *a?G;* its branches determine sequences of *subquestions* for *a?G.* Due to the finiteness of *T* and the form of the rules, only finitely many different subquestions for *OLIG* can be formulated. This together with the requirement not to repeat questions on a branch yields the *Proposition:* The ic-trees for questions *OLIG* are finite, and their branches are closed with either Y or N. This assignment to questions at leaves of an ic-tree can be extended to all questions in the tree and determines a unique value for the original question a?G; the value of a question a*; /?*?<£* is indicated by *[a*; 0*?G*].* In the remainder of this section we will show: if Y is assigned to the root of the ic-tree, then there is a p-normal proof leading from the assumptions to the goal of the question. In the next section this fact will be complemented by a second fact: if N is assigned to the root of the ic-tree, then there is not only no p-normal proof, but no proof at all; i.e., the ic-tree contains enough information to show that the inference from a to *G* is semantically invalid. We will also show that a certain restricted calculus $IC_0(X)$ is still complete; nd-proofs obtained from "derivations" in that calculus are actually normal.

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 we define first the notion of an *ic-derivation.*

Definition. An *ic-derivation for the question* a; /??£? is a subtree T of the ic-tree E for a; /??G satisfying: (i) a; *P?G* is the root of T, (ii) all branches of T are Y-closed branches of E, and (iii) every question node in T (that is not a leaf) is followed by exactly one rule node (to obtain the next question(s)).

One can easily extract ic-derivations from ic-trees that evaluate to Y. Let E be the ic-tree for *a?G* and assume that [<*?G] = Y. We can determine from E a canonical Y-subtree T as *f(hg(E)),* where *hg(E)* is the height of E and / a function defined recursively as follows:

/(0) = *a?G*

to -n — / ^{c}i(/(^{2n})) *f^{some} branch of /(2n) can be extended
f(2n+l) - j $_{/(2n)}$ otherwise

$$f(2n+2) \quad = \quad \begin{cases} \epsilon_2(f(2n)) & \text{if } f(2n+1) \neq f(2n) \\ f(2n) & \text{otherwise} \end{cases}$$

f extends the open branches of a partial ic-derivation by their "left-most **Y**-expansions" in Σ. More explicitly, the open branches of $f(2n)$ are open branches of Σ and are consequently expanded by ic-rules; at least one of these rules must have a (pair of) premise(s) evaluating to **Y**; ϵ_1 chooses the left-most such rule application in each case, and ϵ_2 expands the tree by the appropriate question node(s). The main point is that from an ic-derivation we can construct uniquely an nd-proof[12]; indeed, that proof is p-normal.

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

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

$\wedge_i \downarrow$: The immediate subderivation T_i of Σ has root $\alpha; \beta, \phi_i?G$; by induction hypothesis there is a uniquely determined p-normal nd-proof Π_{T_i} leading from assumptions in $\alpha\beta, \phi_i$ to G. If Π_{T_i} contains occurrences of ϕ_i as open assumptions, then replace those occurrences by $\dfrac{\phi_1 \wedge \phi_2}{\phi_i}$. The resulting p-normal proof of G from $\alpha\beta$ is the associated nd-proof Π_Σ.

$\vee \downarrow$: The immediate subderivations T_i of Σ have roots $\alpha, \phi_i; \beta?G$ for $i =$ 1 or 2; by induction hypothesis there are uniquely determined p-normal nd-proofs Π_{T_i} leading from α, ϕ_i, β to G. The associated p-normal nd-proof Π_Σ of G from $\alpha\beta$ is:

$$\frac{\phi_1 \vee \phi_2 \quad \begin{array}{c}[\phi_1]\\ \vdots \\ G\end{array} \quad \begin{array}{c}[\phi_2]\\ \vdots \\ G\end{array}}{G}$$

This construction is proper, as $\vee \downarrow$ has as its major premise an element of $\alpha\beta$, and G is the endformula of Π_{T_i}.

---

[12] An analogous procedure for the sequent calculus is outlined roughly by Prawitz (p. 91); however, note that no "choices" have to be made in our procedure.

->*l:* The immediate subderivations Ti and T_2 of E have roots a; /3?0i and a; /3, 02?G; by induction hypothesis there are uniquely determined p-normal nd-proofe n_{T1} and n_{T2} leading from a/3 to 0i, respectively from a/3,02 to *G.* Use IIT! and the fact that 0i —> *fa* G a/3 to construct a p-normal proof II of 02 from assumptions in a/3.

$$\begin{array}{c}\vdots\\ \underline{\;01 \quad 01 -> 02\;}\\ \phi_2\end{array}$$

If n_{T2} contains any occurrences of 02 as open assumptions, then replace those occurrences by II. This construction yields the p-normal proof IIE of *G* from assumptions in a/3.

A t: The immediate subderivations T» of E have roots a; /3?0», for *i* = 1 or 2, and *G* is (0i A 02); by induction hypothesis there are uniquely determined p-normal nd-proofe 11^ leading from *a/3* to 0_t. The nd-proof IIE is obtained by joining Ilxi and IIT_2 via A-introduction.

V* t The immediate subderivation T* of E has root a; /3?0j and *G* is (0i V 02); by induction hypothesis there is a uniquely determined p-normal nd-proof $n_{T<}$ leading from *a/3* to 0*. The p-normal nd-proof n_E is obtained by V-introduction.

→↑: The immediate subderivation T of E has root a,0i;/3?02 and *G* is (0i -> 02); by induction hypothesis there is a uniquely determined p-normal nd-proof IIT leading from a, 0i,/3 to 02- The nd-proof IIE is obtained by -^-introduction with 0i and 02-

Finally, we treat the rules for negation.

±*i:* The immediate subderivation T [T^,] of E has root a, *ip;* /??H<p, where *G* is -»V ^d V G J^7; by induction hypothesis there are uniquely determined nd-proofe n_T and IL^ leading from a, *ip, ft* to *(p,* respectively -«p. The nd-proof IIE is obtained by applying ±\ to infer *G.*—The classical rule J_c is treated in the same way as J_i. •

The nd-proof IIE uses exactly the same rules as E. (One parenthetical remark is appropriate here: the structural similarity between ic-derivations and nd-proofs is even more apparent, when the latter are represented graphically by

Fitch-diagrams[13]. The ic-derivations can then be viewed as prescriptions for constructing isomorphic Fitch-diagrams.) Joining the proposition and the earlier observation concerning the extraction of ic-derivations from ic-trees we have:

Proof Extraction Theorem. *For any* α *and* G*: If the ic-tree* Σ *for* $\alpha?G$ *evaluates to* **Y***, then a p-normal nd-proof of* G *from assumptions in* α *can be found.*

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

Interpolation Theorem. *For any* α, G*: if* G *is a logical consequence of* α*, then there is an interpolating formula* ϕ *together with p-normal nd-proofs* Π_ϕ *and* $\Pi_{\phi,G}$*, such that* Π_ϕ *leads from* α *to* ϕ*, and* $\Pi_{\phi,G}$ *leads from* ϕ *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 α—the ic-tree for the question $\alpha?G$ evaluates to **Y** and thus contains an ic-derivation answering the question $\alpha?G$.

Proposition. *For any* Σ, α, β, G*: if* Σ *is an ic-derivation for* $\alpha;\beta?G$*, then there is a uniquely determined interpolant* ϕ*, an nd-proof* Π_ϕ *leading from* $\alpha\beta$ *to* ϕ*, and an nd-proof* $\Pi_{\phi,G}$ *leading from* ϕ *to* G*. Furthermore,* Π_ϕ *and* $\Pi_{\phi,G}$ *are p-normal.*

3. Normal Form Theorems for Sentential Logic. By the evaluation of ic-trees we know that a question $\alpha?G$ obtains the value **Y** or **N**. In case the value is **Y** we can determine an associated p-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

---

[13] Prawitz (1965, pp. 98-99) asserts that already Jaśkowski 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.

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.*

Counterexample Extraction Theorem. *For any a and G : If the ic-tree* **II** ***for ct?G evaluates to* N, *then it contains a canonical refutation branch* P *that determines a valuation* v *with* v \|= </> *for all* <f> G *a and* v J£ *G. (That is,* v *is a counterexample to the inference from a to G.)***

Clearly, if the question *otlG* evaluates to N, so does one of the questions a, *G~7<f>* and a, G~?-»0 for each <*j*> G *T(a,* G~), where we define

$$\phi^{-} = \begin{cases} \psi & \text{if } \phi = \neg\psi \\ \neg\phi & \text{otherwise} \end{cases}$$ and *""* ϕ^{+} = (%l> i f <f> = $\neg\neg\psi$ \ <*f*> otherwise

It will be quite direct to see that the following construction leads to a branch P through II if *T(ot,* G~) is non-empty. If this set is empty, a, G~ consists only of sentential letters. The valuation v, defined for sentential letters *P* by v \|= *P* iff *P* occurs in *a^G",* provides a counterexample. If ,F(c*,G~) is not empty, we need a more sophisticated argument and, naturally, some auxiliary definitions.

The finite set,*T(a₇* G~~) for the negation rules can be enumerated (without repetition) by *(Hi)*$_{ieI}$, where / = {i \| 1 $\leq$ *i* $\leq$ n}. Let H_o = G. Define:

$\kappa(\gamma, i)$ = \| A*M* <k<*nAH*$_k$*&7A -*H*$_k$ £ 7) if there is such an H_k
\ *0* otherwise

The sequence of nodes of P* = P*(0),... is defined as follows:

c*o = *a*
Ao = 0
$\lambda_{m+1} = \kappa(\alpha_m, \lambda_m)$
G_m = H_{λ_m} if $[\alpha_m ? H_{\lambda_m}] = \mathbf{N}$
\ -.#$_{Am}$ otherwise
Q^n+l = *OLm,* G^
P*(2m) = $a_m ? G_m$
P*(2m + 1) = (> ^ + 1 if G_m is a negation
I ^*-cjHx*$_{m+i}$ otherwise

Let 1/ be the smallest m with ^+1 = 0. Define P to be P* restricted to {m I m $\leq$ 2i/}. P is the initial segment of some branch in the search tree;

we call the leftmost such branch the canonical refutation branch. Let us illustrate and clarify this construction through Diagram 2 in the Appendix: At each step in selecting the next question 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 the appropriate -L-nile with the contradictory pair *H** and -**H_k* can be evaluated as Y. (In case both are evaluated as N, we choose the leftmost.) The top node of P is <*„?£„. Let *A* = {<£ | <*f*> G a^G"}. The set *A* has important syntactic closure properties and this can be exploited to define a valuation that will serve as a model for a, G~, i.e., a counterexample to *a* \\= *G.* We establish first the closure properties.

Closure Lemma. *For ail formulas ip :*

(i) ^ <E *A* => *<ip~ $ A;*

(ii) ip is a subformula of an element in A =$• $^+$ G *A or r/)"* G *A;*

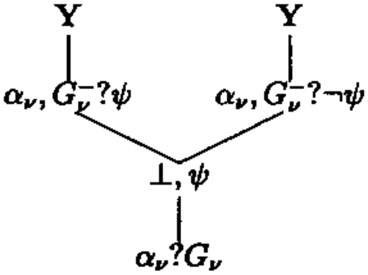
(iii) *i/> is* - " ^ i, *~*-*fa* G *A* => *fa* G *A;*

(iv) ^ (^ A *fa), (fa* A *fa)* G *A* => *<f>t* G *A and <f>%* G *A;*
**l> is ~*(fa b^fa), -»(#!* A *fa)* G *A =* fo* G *A or fe* G *A;*

(v) *i> is (faV fa), (favfa)eA=> <f>t eAor</4* G *A;*
%l) is -»(#! V *fa), -*(fa \\Jfa)eA=^4>i^A and <fc* G *A;*

(vi) *rp is (fa -+ fa), (fa -+fa)eA=><f>ieAor<f4eA;*
tj) is ~(fa—* fa), ~^(fa—> fa)* G *A* ==^ *<f>i* G *A and* < ^ 6 i .

PROOF. We assume for simplicity that *G~ $. ot_u* thus no question node which has *a_v, G~* on the left-hand side will repeat a question in P. (If *G~* G *a_v* and *OL_V*, G~?G* repeats a question *otmiG** on P, then the arguments below are carried out for that earlier question.) (i) Let *tl>* G *A.* If ^ is not a negation and -i^7 G -A then the following subtree is in the search space:



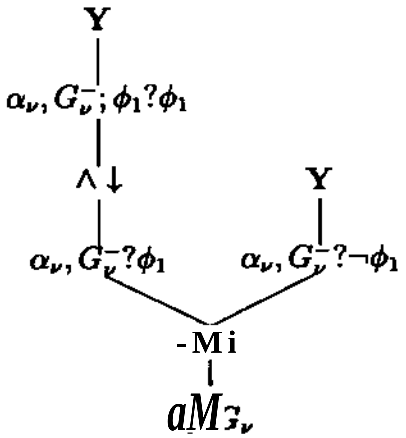
Thus [a,/?G_v] = Y, contradicting the construction of P. If *rp* is a negation, the argument proceeds similarly.

For (ii), let <*j*> G *A* and *rp* a subfonnula of <*f*>. Assume as case 1, that *ip** = ^. If *tj*> = <*f*>, we are clearly done; so suppose *tp* is a proper subfonnula of <*f*>. Then either ^ or *rj)~* is an element of *Tfa^G')* = ^(ajG"). If ^ G a,G^ or ^~ € a,G~, we are done. Otherwise some G_m, m > 0, is one of Vs-'V^","^{1}^"")- <2~ is one of ^~, ("•VO"* V^r~~> (~^{I}(V'~))~- These are, respectively, ^""$_L$^,^, ^'"> as we are supposing V>$^+$ = ^- Thus, G~ is either *rl)*+ or V^~, and as* G~ € A, either *ip* £ A OT ip~ € A.*

Assume as case 2 now that *%l)*$^+$ ^ *xj),* i.e., ^ = -I-IX f$^{\circ r}$ sonie x» a^{11}^ ^$^+$ = x- Then -ix is an element of *T{OL$_V$, G~)* = ^(or, G~). ^ neither -ix ^{nor} X is in a, then some G_m, rn > 0, is either -ix or -»-ix- ^m̄ ^ then either x or -ix; the former is ^$^+$, the latter *i[>~*. Thus, as before, *tp*+ G -A or ^r G -A.

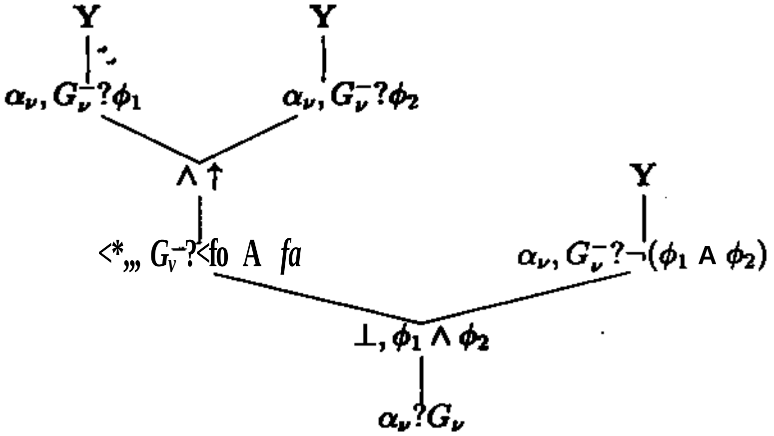
For (iii) assume ~~i i~~\$_x$ G A. By (ii) either (->-»0i)$^+$ G A or (-«-«^i)" G *A]* by (i) the latter case cannot arise. Thus (~»"^{i}^i)$^+$ G A; ("»-'0i)$^+$ is <£i and we can conclude that *4>i € A.*

The arguments for the remaining items are similar. We present the argument only for (iv). First let (0i A *fa)* G *A* and assume <\>\ *£ A* (the case <*f*>*2 & A* is symmetric); by (ii) <*f*>\ € A- If #i is^{not} a negation, then <£j~ = $\neg\phi_1$, and the following subtree is in the search space:

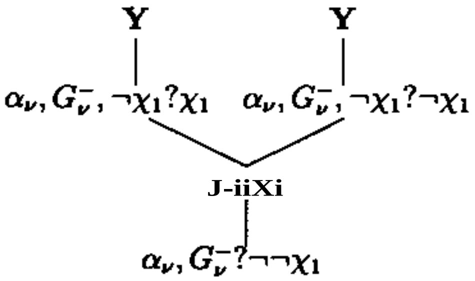


so $[a_u?G_v]$ = Y which contradicts the construction of P. If *4>* is a negation, then we have a symmetric tree which again yields a contradiction.

Here the application of A 4- is crucial. To establish the second part of (iv) A t is used analogously. So assume that ->(^i A ^) e A ^ " ^ *A,* and *fe & A.* By (ii) we have that *4>t € A* and *4$* G *A.* For simplicity's sake let us first consider the case that *<f>f* =.&. Then we have:



If 0i is -"Xi and 0j is 02, then the left branch over the question node A t has to be replaced by



The remaining cases 0† is 0i, but 02 is -»-*X2, and 0» is ->-»Xt ^*^{e}* treated similarly. D

Now define a valuation by v f= *P iff P* G *A*. Using this valuation and the closure lemma we can prove the proposition: for every 0 G *A* , v f= 0. Hence v is a model for a,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 a?G allows us to determine either a p-normal proof of G from a or a branch that provides a counterexample to the inference from a to G.*

This yields, a *semantic proof of the p-normal form theorem* for the natural **deduction calculus.**

P-Normal Form Theorem. *If G can be proved from assumptions in a, then there is a p-normal proof of G from a.*

As a matter of fact, the proof establishes more, as the nd-proofs obtainable from ic-derivations are a proper subclass of p-normal derivations; for example, the following derivations

$$\dfrac{\dfrac{[\phi_1] \quad \phi_1 \to \phi_2}{\phi_2}}{\text{01 –> 02}}$$

and

$$\dfrac{\dfrac{\text{01 A 02}}{\text{01}} \qquad \dfrac{\text{01 A 02}}{\text{02}}}{\text{01A02}}$$

are p-normal, but not obtainable from an ic-derivation. Notice that these derivations are actually normal and that one can construct such derivations of arbitrary length. As a matter of fact, the "normal" form can be further restricted. But before considering such additional restrictions we would like to re-emphasize one absolutely central point: the normality of the nd-proofs obtained from ic-derivations is a direct consequence of (the very intuitive strategy for constructing nd-proofs that underlies) the generation of ic-trees for particular questions. That intuitive strategy consists of trying to close the gap between assumptions and conclusion "from above" (by elimination rules) and "from below" (by inverted introduction rules); if neither works, one proceeds indirectly. Thus, a 4-rule can only be applied to assumptions or to formulas that have been inferred by 4-rule applications; similarly, the conclusion of $\pm_c$ cannot be the major premise of a proper elimination rule.

Further restrictions on "normal" forms are obtained by restricting the generation of ic-trees; we discuss this here only for modifications of the ±-rules. This will lead to normal nd-proofs. In the above discussion we considered J-c essentially as an I-rule for complex non-negated formulas; to a formula thus introduced no E-rule can be applied. Why not consider also negated formulas and disallow subsequent applications of $\pm_c$ (now viewed as an E-rule)? That excludes then in particular nd-proofs of the form

$$\dfrac{\begin{matrix}[\neg\psi]\\ \vdots\\ \phi\end{matrix} \qquad \dfrac{\begin{matrix}\vdots\\ \chi\end{matrix}\quad\begin{matrix}\vdots\\ \neg\chi\end{matrix}}{\neg\phi}}{\psi}$$

Indeed, this is just a special case: in a normal proof, no major premise of a -L-rule is the conclusion of a _L-rule. That the ic-calculus can be restricted in such a way as to provide only normal nd-proofs (without loss of completeness) will be a consequence of the subsequent considerations.

For proof search it is important that ic-trees be pruned—without losing completeness. That can be achieved by restricting the formulas with which contradictory pairs are formed; one can do this through four successively more restrictive versions of the operation ^(7), namely, *Af(y),* ^(7)1 £(7)> and X(7). iV_7 (P_7, S_7) consists of all negations that occur as (positive, strictly positive)[14] subformulas in 7; J_7 contains exactly the elements of 7 that are

---

[14] These notions are defined in the Appendix.

negations. ^(7) (^(7), 5(7), *I(j))* consists then of the formulas *tp* with ¬ψ in N_7 (P_7, 5_7, $/_7$). The ±-niles for these operations are now formulated, except for X, as indicated earlier; the ±(J)-rules are given as follows:

±$_C$(J): a; /??<?, *tp e* J(a£, -.G) =*• a, -.G; /??<? AND a, -.G; 0?-.^

±i(I) is given in a similar way. Clearly, these rules can be reformulated as

$\perp_c(\mathcal{I})$: a; /??G, <*p* G J(a£, -G) => a, -*G*; β?φ

and similarly for -U(X); this brings out most clearly that *-yip* is "immediately available".[15]

To establish completeness for each of the resulting variations of the ic-calculus, it suffices to show that the restricted ic-trees (built up by the |- and f-niles, and the restricted -L-riiles) allow the extraction of a counterexample in case [a?GJ = N. The construction of a canonical refutation branch involves now not only the ±-rules, but possibly 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. Prom this fact for IC(I) we can infer the normal form theorem below: the adjacency condition is obviously satisfied in this case, as the major premise a, G~"; /??-»¥> of the ±-rules has an immediate Y-answer.

Normal Form Theorem. *IfG can be proved from assumptions in a, then there is a normal proof of G from* a.

Remark. Before extending our considerations to full predicate logic, let us return to some general remarks we made in section 1. There we emphasized the role of the ic-calculus as a technical tool in the search for nd-proofs. The rules are directly modeled after the I-, E-, and -L-rules of the classical natural deduction calculus (with a special treatment of classical negation). However, due to the way in which assumptions are indicated and ^-niles are represented, there is also a certain resemblance with the sequent calculus.[16]

Two distinctive features of the ic-calculus were already mentioned in note 10 and remark 2 at the end of section 1. Here we note some additional (and obvious) differences with the sequent calculus: (i) the ic-calculus always has

---

[15] A trivial modification is now needed in the proof extraction lemma.

[16] If it were just for the first feature, we would have essentially the formulation of NK as given in [Gentzen 1936], pp. 512-515.

exactly one formula on the right-hand side; (ii) every formula on the left-hand side of a conclusion appears on the left-hand side of the premise(s); (iii) redundant formulas may not be inferred on the left-hand side; (iv) the negation rules have been altered. To put it briefly and informally: the ic-calculus is a special form of natural deduction, where the goal is never left out of sight!

It has been suggested that the sequent calculus could be used as well as the ic-calculus in the search for nd-proofs, i.e., one would proceed in two steps:

(i) search for a proof in, say, Gentzen's LK (or alternatively in a tableau system, which can be viewed as a notational variant of LK);

(ii) translate the resulting proof into an nd-proof.

In (i), LK may be restricted so as to make for more efficient search, and, in addition, allow an easier translation to NK or provide more natural NK proofs. In (ii), the LK proof itself may be manipulated before translating to achieve the additional goals just mentioned. (We referred to work along these lines in note 8, in particular that of Shanin e.a.) From this point of view one might look at the technical aspects of our paper as imposing particular restrictions on LK which make search more efficient and translation into NK trivial.

However, this view is rather forced: it brushes aside not only all differing "details" of the calculi, but also the strategic use of the ic-calculus for building up an *appropriate* search space. The search space should be appropriate for our main goal, i.e., it should allow us, from the very beginning, to focus on the question of *finding* "natural" NK proofs by using "natural" search strategies.[17] The most obvious of these strategies is to work backward from the goal formula and forward from derived lines, both in a restricted and goal-directed manner, i.e., to perform sequences of intercalation steps. We try to find simple representations of the states of the search and of the transition steps taking us from one state to the next. The representation of the final state of a successful search must encode a "proof that can be directly viewed as an NK proof. This difference in the strategic use of the ic-calculus comes out clearly in the completeness proof for the calculus presented in this section (and was emphatically stated in section 1).

---

[17] Indeed, this opens interesting questions for proof theoretic study, e.g., how is the form of nd-proofs related to the strategies used in their search?

4. Normal Form Theorems for Predicate Logic. The metamathematical considerations for sentential logic can be extended to predicate logic. To that end we use the following formulation of the E- and I-rules for the quantifiers; note that writing ϕt assumes that t is free for x in ϕx or, alternatively, that some bound variables in ϕx have been renamed. For $\forall$ we have the rules:

$$\frac{(\forall x)\phi x}{\phi t}\ \forall\mathrm{E} \qquad\qquad \frac{\phi y}{(\forall x)\phi x}\ \forall\mathrm{I}$$

Applications of the I-rule must satisfy the restriction that y does not have a free occurrence in any assumption on which the derivation of ϕy depends.— For $\exists$ we have the rules:

$$\frac{(\exists x)\phi x \qquad \begin{matrix}[\phi y]\\ \vdots \\ \eta\end{matrix}}{\eta}\ \exists\mathrm{E} \qquad\qquad \frac{\phi t}{(\exists x)\phi x}\ \exists\mathrm{I}$$

with the usual restriction on the E-rule, namely, y must not have free occurrences in η or $(\exists x)\phi x$ nor in any assumption (other than ϕy) on which the proof of (the upper occurrence of) η 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 $\mathcal{T}(\gamma)$ denotes the finite set of terms occurring in the formulas of γ.[18]

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

$\exists\downarrow$: $\alpha;\beta?G$, $(\exists x)\phi x \in \alpha\beta$, y is new for α, $(\exists x)\phi x, G \Longrightarrow \alpha,\phi y;\beta?G$

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

$\exists\uparrow$: $\alpha;\beta?(\exists x)\phi x$, $t \in \mathcal{T}(\alpha\beta, (\exists x)\phi x) \Longrightarrow \alpha;\beta?\phi t$

Ic-trees are specified inductively: if $\alpha^*;\beta^*?G^*$ is an open question, all possibilities of intercalating formulas are considered as in the case of sentential logic. Let us just remark that for applications of the $\perp$-rules we are considering as (proper) subformulas of quantified formulas all instances with the

---

[18] As in the propositional calculus, we add restrictions to the $\downarrow$-rules which prune the search space. In the case of $\exists\downarrow$, the restriction is that there is no t such that $\phi t \in \alpha\beta$; in the case of $\forall\downarrow$, we require that $\phi t \notin \alpha\beta$.

finitely many terms in $\mathcal{T}$. The resulting calculus is denoted by $\mathrm{IC}_1(\mathcal{O})$, depending on the set of formulas admitted for the $\perp$-rules. 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 [p. 334] for three-valued logic: If N is a leaf of Σ, $[N]_\Sigma = \mathbf{v}_\Sigma(N)$, and in case N is the unique successor of M, $[N]_\Sigma = [M]_\Sigma$. In case N is at a conjunctive branching,

$$[N]_\Sigma = \begin{cases} \mathbf{Y} & \text{if for all immediate predecessors } M \text{ of } N\text{: } [M]_\Sigma = \mathbf{Y} \\ \mathbf{N} & \text{if for some immediate predecessor } M \text{ of } N\text{: } [M]_\Sigma = \mathbf{N} \\ \mathbf{U} & \text{otherwise} \end{cases}$$

and in case N is at a disjunctive branching,

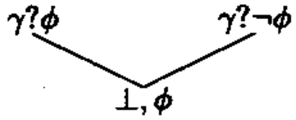
$$[N]_\Sigma = \begin{cases} \mathbf{N} & \text{if for all immediate predecessors } M \text{ of } N\text{: } [M]_\Sigma = \mathbf{N} \\ \mathbf{Y} & \text{if for some immediate predecessor } M \text{ of } N\text{: } [M]_\Sigma = \mathbf{Y} \\ \mathbf{U} & \text{otherwise} \end{cases}$$

The full ic-tree Σ for $\alpha?G$ is defined in stages as follows: Σ_0 is $\alpha?G$; Σ_{n+1} is Σ_n if $[\alpha?G]_{\Sigma_n}$ is either **Y** or **N**, otherwise Σ_{n+1} is obtained from Σ_n by expanding each open branch by all applicable rules. Three possibilities can arise: (1) for some $n \in N$, $[\alpha?G]_{\Sigma_n} = \mathbf{Y}$, (2) for some $n \in N$, $[\alpha?G]_{\Sigma_n} = \mathbf{N}$, and (3) for all $n \in N$, $[\alpha?G]_{\Sigma_n} = \mathbf{U}$. In the first case a p-normal derivation can be associated with a subtree of Σ_n—by selecting an ic-derivation and by proving (inductively) that each ic-derivation determines a unique p-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.

Counterexample Extraction Theorem. *For any α and G: if the ic-tree Σ for $\alpha?G$ is such that for every natural number n $[\alpha?G]_{\Sigma_n} = \mathbf{U}$, then Σ contains an infinite refutation branch* P *that determines a structure $\mathcal{M}$ with $\mathcal{M} \models \phi$, for all $\phi \in \alpha$, and $\mathcal{M} \models \neg G$. Thus, $\mathcal{M}$ is a counterexample to the inference from α to G.*

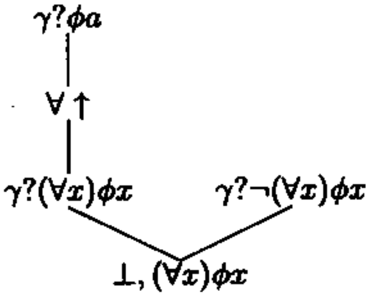
The extraction of a counterexample from an infinite ic-tree requires some circumspection: Instead of constructing a refutation branch directly, we determine first a particular infinite subtree E* of the ic-tree E and then apply König's Lemma to this *canonical refutation tree.* The reason for having to cut down the ic-tree E to the canonical refutation tree E* is this: Refutation branches have to satisfy suitable closure conditions, and it is trivial to construct infinite branches of E that don't. So we define E* 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" tableau constructions; thus we emphasize only the crucial points.

The construction of E* (as a subtree of the ic-tree E) for the question *a?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 E* (using conventions and definitions from the sentential case) with the first wave for the enumeration of the formulas in *T{a, G~)* as in the sentential logical case with E*(0) = a?G. For *m* > 0, we extend each open branch of E*(2m) (i.e., its leaf evaluates to U) with a rule node of theipnn -L, ^



if both questions *y?tf>* and 7?-»0 evaluate to U; if only one of them evaluates to U, then the branch is extended at just that question. One of these cases must hold, because the rule node JL, <*j*> has value U. (Clearly, as before, <*f*> is the first element in the given enumeration that extends 7 properly.) 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 the JL-rules, as for all n G *N* [a?G]s:$_n$ = U. In sentential logic, we saw, that cannot happen; the resulting set of formulas *A* is deductively closed in the sense of the earlier Closure Lemma. Here, some of the *A*^{J}*s* associated with leaves cannot satisfy the closure conditions *(3x)<f>x* G *A* => *</)+t* G *A*

for some term t, and -«(Vx)0x G *A* ==> *(f>~t* G *A* for some term *t.* In the first case the rule 3 *I* is applicable with a canonically chosen new variable; in the second case we are able to extend the branch in the following way using also 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 3 ± in all needed cases and then perform the above analysis on those -»(Vx)0x 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 subtree E* of the ic-tree; Kõnig's Lemma applied to E* yields an infinite branch P. Define *Ap* = *{ifi \ y\>* 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{Ap)* consist of all terms that occur in some formula of A_P.

Closure Lemma. *For all formulas rj):*

(i) *il)* G *Ap* = * *rl>-* <£ *Ap;*

(ii) t/> is a subformula of an element in Ap => ^+ G *Ap or tp"* G A_P;

(iii) *rf) is* -1-^1, "»-»0i G *Ap* => *fa* G *Ap;*

(iv) V *is* (0i A *fa), {fa Afa)eAp=><f>teAp and* <££ G A_P;
rj) is -y(fa A *fa), -^(fa Afa)e*A_P => <f" G A_P *or fe* G A_P;

(v) *tl>is(faV fa), {fa Vfa)eAp=>4>teApor<t4e* A_P;
rp is -{fa* V *fa),* -i(0! V *fa)* G A_P ==> <£j' G *Ap and fe* G *Ap;*

(vi) ^ *is {fa -> fa), {fa -+fa)eAp=>4>ieApor<f4e Ap;*
ip is -->{fa -» fa), ->{fa -> fa) G *Ap* => *<f>f* G *Ap and fe* G A_P

(vii) *rp is (3x)<f>x, (3x)<l>x* € A_P => *<f>+t e* A_P *for some term t e T(*A_P*);*
ip is -i(3x)<f>x, -^(3x)(f)x e A_P => *(j>~t e* A_P *for all terms t e T{*A_P*);*

(viii) *xp is* (Vz)<£z, (Vx)0x G A_P => <\/>+* *e* A_P *for all terms t* G *T(*A_F*);*
%l) is ->(Vx)<f>x, -»(Vx)<^x G A_F => *<f>~t* G A_P *for some term t* G *T(*A_P*).*

The definition of a structure *M.* from *Ap* 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 determines either a p-normal nd-proof of G from a or a branch that provides a counterexample Ai to the inference from a to G.*

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

P-Normal Form Theorem. *If G can be proved from assumptions in a, then there is a p-normal nd-proof of G from a.*

The .L-rules can be restricted to smaller classes of formulas; that provides then, as in the case of sentential logic, the argument for the normal form theorem.

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

If we were just concerned with establishing normal form theorems, we could end our paper right here. However, we want to provide the broad theoretical basis for proof search in first order logic. That requires additional work, namely, to find the basis for a natural extension of the search algorithm for sentential logic, as implemented in the Carnegie Mellon Proof TVitor.[19]

5. **Skolem-Herbrand Expansion.** For the search algorithm the language of predicate logic is expanded by *new free variables* and *Skolem and Herbrand*

---

[19] Quite sophisticated strategies are involved in the algorithm underlying the Proof Tutor that searches automatically for nd-proofs in classical sentential logic; that program was developed by Richard Scheines and Wilfried Sieg with assistance from Jonathan Pressler and Chris Walton. Presently we are redesigning it in collaboration with Jesse Hughes, Mark Ravaglia, Richard Scheines, and Prank Wimberly, and we have extended the search algorithm to predicate logic along the lines sketched here.—Similarly motivated programs have been developed by Jeff Pelletier and Fred Portoraro; cf. their papers in this volume.

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 "strictly positive canonical subfonnula" and a unification algorithm for quantified formulas, see [Sieg and KauSmann]. We will come back to these issues at the end of our paper, briefly. Here we focus on the description of the search space, i.e., the generation of appropriate ic-trees.

C is the language of ICi; the *Skolem-Herbrand expansion* ICSH has as its underlying language an expansion £SH of *C\\ C* is fixed here to have just the set *X* = {x, x_0, x_u...} as its set of variables. £SH has in addition a set *Y* of bound variables {y, j/o> ! / i, . . . }, a set Z of parameters {z, *ZQ*, *Z*, . . . }, and a set *F* of function symbols {/, $/_0$, /i, •. . }. The sets *X*, *Y*, and *Z* are all disjoint; *F* contains infinitely many function symbols for each arity n, n a natural number; the 0-ary symbols are constants. Terms and formulas of $£_S$H are inductively generated as usual. Let us just note that we call a given variable or function symbol *new* for a given tree (or set) if that symbol does not occur in the tree (or set). For a sequence of formulas 7, 7SH(7) is the set of terms in 7; 71(7) contains exactly those terms in 7 which are terms in $£_{i;}$ T*($_7$) = 7$£_H$(7) - 71(7). Similarly we use £* = $£_S$H - A- Finally, the set of parameters'of 7 is given by FV(7) = 7SH(7) n *Z*. The ICSH calculus is obtained from ICi by replacing the quantifier rules with those which appear below:[20]

V*1:* a; /??G, (Vz)<£z € a/3 => a; /?, *<f>z?G* for some new *z*

3 I: a; /??G, *{3x)<f>x ea@,z̄* = FV(a, (3x)0x, *G)* => a, *<t>f(z̄); P?G* for some new /

V t: a;)8?(Vx)0x, z̄ = FV(a, (Vx)0x) =* a; */3?<f>f(z̄)* for some new /

3 t* <*; */3?(3x)(f>x* =» a; *p?<j>z* for some new *z*

Parameters and function symbols are new relative to (partial) ic_{SH}-trees. Such trees are built up in the most straightforward way by using the rules of

---

[20] We can, of course, make the same kind of restrictions as before concerning inference of repeated formulas; cf. note 18. We can also improve efficiency by restricting ourselves to canonically chosen function symbols (one for each formula up to renaming of variables) and by taking as parameters for the term only the "relevant" variables—both strategies are discussed and analyzed for tableaux by Baaz and Fermüller, 1995.

IC_{SH}; what is not as straightforward is the formulation of appropriate closure conditions. I.e., branches will be closed with Y, N, and U under roughly the same conditions as before, but now we consider also "partial" yes-answers *Y_a* relative to a "unifying substitution" *a.* The reason is simple, as the question "a; *0?G"* is now asking "Is *G* unifiable with an element in a/??" In case we find unifying substitutions, we close the branch with a sequence of Y^'s and, in case other rules can be applied, also with U. In the last case, all other options of intercalating formulas are used to expand the partial icsH-tree.

Three points have to be taken up: (1) appropriate unification, i.e., a substitution concept generalized to formulas; (2) evaluation of partial icsH-trees that uses the unification information properly; (3) extraction of nd-proofs from icsH-trees. The last issue and normal form theorems will be addressed in the next section. (2) will be quite naturally resolved, as soon as (1) is properly set up.

Definition. A *term assignment* is a mapping *a* from *Z* to the terms of £SH such that sup(a) = *{z \ a(z) ^ z}* is finite. If sup(a) = {zb,..., *Zn}* then *a* can be represented by (a(zb)/zb,..., *<r(zn)/zn);* () = id_z.

Substitutions, based on term assignments, will include a canonical renaming of bound variables. For that we have to consider "modifications" of term assignments ^(W.·· •:*»/«>«) for variables two,..., *w_n* € *X* U *Z* and terms *o> • • •»*n € *Csw* The modification is given for *w* € *X* U *Z* by**

$$\sigma^{(t_0/w_0,\ldots,t_n/w_n)}(w) = \begin{cases} ti & \text{if tu = W\{ for some } i \leq TI \\ \text{^"(w)} & \text{otherwise} \end{cases}$$

Note that this will not be, in general, a term assignment, as variables from *X* may appear in the support. For a given modified term assignment *a* we define a family of (i, ^-substitutions on *X* U *Z* as follows: a»[x] = a(x) and *Gi[z]* = a(z) in the base case; *Oi* distributes over function and relation symbols, but also over the sentential connectives. For quantified formulas *{Qw)<f>*, where *Q* is V or 3,

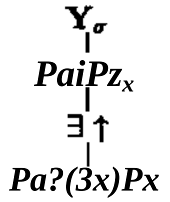
$$\sigma_i[(Qw)\phi] = (Qy_i)\sigma_{i+1}^{(y_i/w)}[\phi].$$

For a given term assignment a, we write *a* for *&Q* and call any (i, a)-substitution simply a substitution.

Notice that *<T[(VXI)(3X_2)PXIX_2]* is (Vy_o)(3i/i)Pyoyi; applying *a* to (Vx_7) *(3x_3)Px_7x_3* yields the same result; i.e., *a* literally identifies formulas that

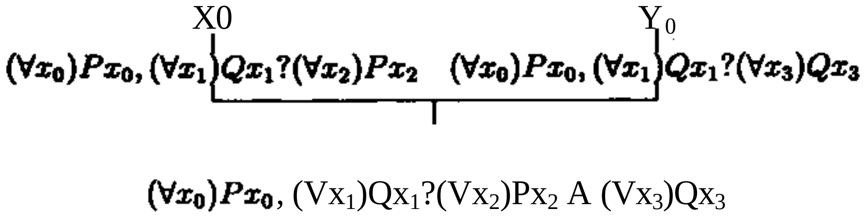
are identical only up to renaming of bound variables. It is the canonical renaming of bound variables that allows us to extend unifiability from terms to formulas: Two formulas <*t*> and *tp* in £SH are called *unifiable* iff there is a term assignment a, such that *o[<j>]* = cr[^].[21] Let us look at some examples on how to prove statements in the expanded calculus and motivate the additional technical steps we have to take.

Example 1. *Pa* h *(3x)Px*



a is the substitution (a/zi); as example 3 will show, closing with Y_a will not always guarantee success of the proof search.

Example 2. $(yx_o)Px_o$, (Vx^Qxi h $(Vx_2)Px_2$ A $(Vx_3)Qx_3$



This transforms directly into an nd-proof, as we assume the general renaming rule; cf. **beginning** of section 6.

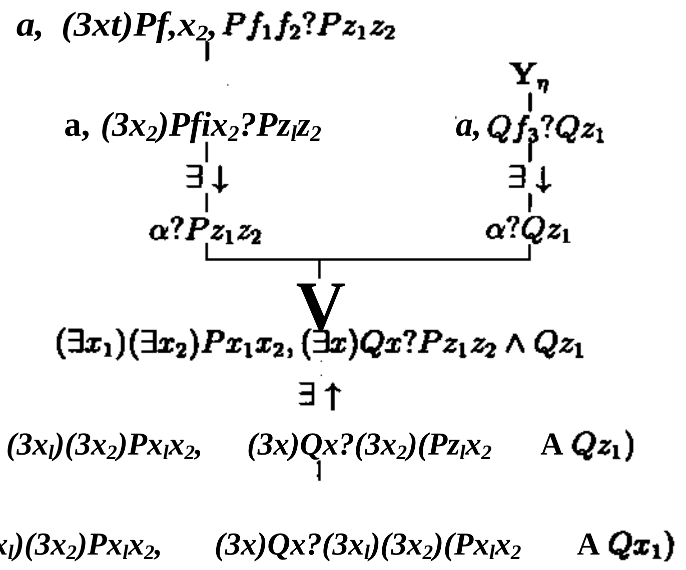
Example 3. Closing every branch with a substitution is not enough to guarantee that a derivation has been found. For example,

$(3x_l)(3x_2)Px_lx_2$, $(3x)Qx$ \f $\{3x_l)\{3x_2)\{Px_lx_2$ **A** $Qx_x)$.

Consider the following partial icsH-tree (where *a* abbreviates the appropriate sequence of assumptions):

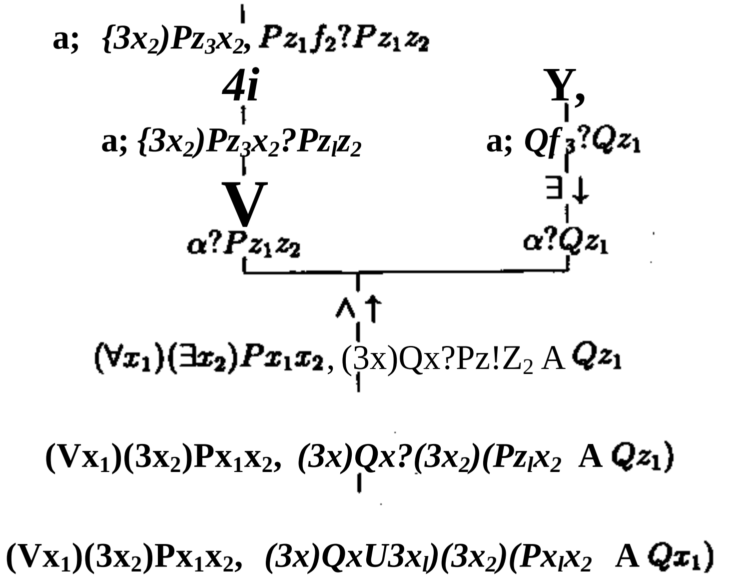
---

[21] **Standard unification algorithms can be easily adapted to provide a most general idempotent unifier; cf. [Sieg and Kauffmann].**



Here, *a* = *(fi/ziifa/z^* and r; = *(fzfz\)*. *a* unifies P/1/2 with *Pz\z*$_2$ and ?/ unifies *Qfz* with Qzi, but applying both of these means that P/1/2 and *Qfz* will be the premises to A t with a conclusion which should unify with *Pz\z*$_2$*t\Qz*. The only possible conclusion from these premises is P/1/2 AQ/$_3$, which does not unify with *Pz\Z2* A *Qz*. Thus the failure of this tree to be a derivation is not determined at the leaves, but rather when the unification information is passed down the tree. Note also that in applying 3 *X* to obtain Q/35 we introduce a function symbol which is new to the entire tree, not just to the branch below the rule application. If we required only that these names were new to the given branch, we could have instantiated *(3x)Qx* with *Qf*. Doing so would have allowed us to use *rf* = *(fi/zi)*. The tree would then be a "derivation".

Example 4. ***(Vxi)(3x***$_2$***)Px***$_x$***x***$_2$***,{3x)Qx* h *(3XI)(3X*$_2$*)(PJ:*$_1$*X*$_2$ A *Qx*$_x$*)* The derivation is as follows:**



Here a is *(z&/zi, h/z^*, and r/ is *(fz/z$_x$)*. In this tree, when passing down the unifier **informatfen,** we can "merge" *o* and *rj* to a substitution (/3/21, /a/^, *hl*i)-* This substitution will work for the part of the tree where the two branches merge.

The following considerations serve to make explicit the mechanism underlying the "merging" mentioned in example 4. For that purpose we review first some standard definitions as found, for example, in [Snyder]. We indicate the composition of two substitutions *a* and *p* by *ap,* with *ap(w) = a(p(w)).* A substitution *a* is t *dempotent* just in case *GO = a.* Finally, *p<o* ("p is more general **than** a" **or, perhaps better, "p** is less specific than a") if and only if there is a substitution 77, such that *r)p = a.* For idempotent substitutions we note that *p* ≤ *a* implies *op = o.* Finally, we come to our crucial definitions.

Definition, (i) For substitutions *O* and *o₂, o = O\\/o₂* is the least (with respect to ≤) substitution *o* such that *o* ≤ *o* and *o₂* ≤ *o. o* is called the *join* of *o* and *o₂.* Note that the join is not always defined, and that sometimes multiple substitutions may be joins for a given pair of substitutions (in which case we simply pick one), (ii) Substitutions *o* and *p* are called *consistent* exactly when *o\lp* is defined.

Let us consider two examples. If $\sigma = \langle a/z_1, b/z_2\rangle$ and $\eta = \langle c/z_1\rangle$, then $\sigma[Pz_1z_2 \wedge Qz_3] = Pab \wedge Qz_3$ and $\eta[Pz_1z_2 \wedge Qz_3] = Pcz_2 \wedge Qz_3$, whereas $\sigma\dot{\vee}\eta$ is undefined. Now let $\sigma = \langle z_3/z_1, b/z_2\rangle$ and $\eta = \langle c/z_1\rangle$, then $\sigma[Pz_1z_2 \wedge Qz_3] = Pz_3b \wedge Qz_3$ and $\eta[Pz_1z_2 \wedge Qz_3] = Pcz_2 \wedge Qz_3$. In this case $\sigma\dot{\vee}\eta = \langle c/z_1, b/z_2, c/z_3\rangle$ and $\sigma\dot{\vee}\eta[Pz_1z_2 \wedge Qz_3] = Pcb \wedge Qc$.

To summarize our discussion (through examples): When asking a question in the calculi described prior to $\mathrm{IC_{SH}}$, closing a branch with **Y** guaranteed success along that branch, and succeeding for the whole proof required only that we succeed on sufficiently many branches to build a derivation. The above examples illustrate that closing off a branch with a unifier does not guarantee success on that branch, as it may cause the new free variables occurring in the branch to be instantiated with terms which are not consistent with the rest of the tree. Thus, a unifier gives us success only modulo its compatability with unifiers from other branches of the tree. Moreover, at any stage there may be multiple possible unifiers, any of which may or may not succeed further down the tree.

In order to keep track of all of these possibilities we modify the valuation function accordingly. We will introduce the value $\mathbf{Y}_\sigma$ for every idempotent substitution σ. Roughly speaking, a node N will be given the value $\mathbf{Y}_\sigma$ if applying σ to the subtree rooted at N will result in a tree having value **Y**. To do this rigorously, we first introduce a means for "joining" *values*, so that the value for the whole tree can be determined—when the leaves have values, namely, sets of $\mathbf{Y}_\sigma$'s.

Definition. Let $A = \{\mathbf{Y}_{\rho_1}, \ldots, \mathbf{Y}_{\rho_m}\}$, $B = \{\mathbf{Y}_{\sigma_1}, \ldots, \mathbf{Y}_{\sigma_n}\}$. $A\dot{\vee}B = \{\mathbf{Y}_{\rho_i\dot{\vee}\sigma_j} \mid 1 \le i \le m, 1 \le j \le n,$ and ρ_i and σ_j are consistent$\}$.

The earlier evaluation function $[N]_\Sigma$ has to be modified, as sets of values are assigned to nodes. Let Σ be a partial $\mathrm{ic_{SH}}$-tree and $\mathbf{v}_\Sigma$ the valuation for the leaves of Σ. In case N is a leaf, say $N = \alpha; \beta?G$, $[N]_\Sigma = \{Y_\sigma \mid \sigma[G] \in \sigma[\alpha\beta]$ and σ idempotent$\}$. In case N is the unique successor of M, $[N]_\Sigma = [M]_\Sigma$; in case N is at a conjunctive branching, i.e., a rule node for a two-premise rule with M_1 and M_2 above N, $[N]_\Sigma = [M_1]_\Sigma\dot{\vee}[M_2]_\Sigma$; finally, in case N is at a disjunctive branching, i.e., a question node with rule nodes $M_1, \ldots, M_k$ above N, $[N]_\Sigma = \bigcup_{1\le i\le k}[M_i]_\Sigma$. If Σ has (question node) N as its root, we set $[\Sigma] = [N]_\Sigma$.

6. Correctness of Proof Search (in the SH-Expansion). We consider the SH-expansion "just" as a convenient technical tool for automated proof search; thus, we ask basic questions $\alpha?G$ only when the elements of α and G are formulas in $\mathcal{L}_1$. Clearly, if we find an $\mathrm{ic}_{\mathrm{SH}}$-derivation for such a question, we want to associate with it an nd-proof in $\mathcal{L}_1$ of G from assumptions in α. This is immediate with p-normal or normal derivations, as soon as we know how to transform partial $\mathrm{ic}_{\mathrm{SH}}$-trees into ic_1-trees. For this purpose, we define a *canonical renaming function* as follows:

Definition. Let σ be a substitution and Π an $\mathrm{ic}_{\mathrm{SH}}$-tree. Let $\{t_1, \ldots, t_n\} = \mathcal{T}^*(\sigma[\Pi])$. Let $x_1, \ldots, x_n \in X$ be new for Π. R_σ^Π is the *tree-renaming generated by* Π *and* σ.

$$R_\sigma^\Pi[t] = \begin{cases} x_i & \text{if } \sigma[t] = t_i, 1 \leq i \leq n \\ \sigma[t] & \text{otherwise} \end{cases}$$

R_σ^Π distributes over relation symbols, sentential connectives, and quantifiers. It is applied to a question node by applying it to every formula at the node, and to a tree by applying it to every question node in the tree (as well as to the formulas displayed at $\perp$-rule nodes). We abbreviate $R_\sigma^\Pi[\Pi]$ by $R_\sigma[\Pi]$.

For such renamings we will show that they associate with partial $\mathrm{ic}_{\mathrm{SH}}$-trees partial ic_1-trees. Then we will show that an $\mathrm{ic}_{\mathrm{SH}}$-derivation exists for a given $\mathcal{L}_1$-question $\alpha?G$ if and only if an ic_1-derivation exists for $\alpha?G$. We add to our formulation of the natural deduction calculus a rule that allows for renaming of bound variables:

(R) $\dfrac{\phi}{\phi^*}$ if $\sigma[\phi^*] = \sigma[\phi]$ for every term-assignment σ.

We need to make the corresponding adjustment to IC_1, i.e., we close a branch with leaf node $\alpha; \beta?G$ with **Y** whenever $G \in \alpha\beta$ up to renaming of bound variables.

Local Correctness Lemma. *Let* T *be an* $\mathrm{ic}_{\mathrm{SH}}$*-tree for* $\alpha_0?G_0$—α_0, G_0 *in* $\mathcal{L}_1$. *Then for any substitution* σ, $R_\sigma[\mathrm{T}]$ *is an* ic_1*-tree for* $\alpha_0?G_0$.

PROOF. Fix $R = R_\sigma^{\mathrm{T}}$; by the definition of R, $R[\mathrm{T}]$ has root $\alpha_0?G_0$ (since the support of R is contained in $\mathcal{L}^*$). We show now by induction on the height of Π^* that for every subtree Π^* of T, $R[\Pi^*]$ is an ic_1-tree. Let $\alpha; \beta?G$ be the root of Π and let $\alpha'; \beta'?G' = R[\alpha; \beta?G]$. If $\alpha; \beta?G$ is a leaf node then $R[\alpha; \beta?G]$ is a leaf in $\mathcal{L}_1$ and hence an ic_1-tree. For the inductive step, it will

**be enough to show: for each rule node r immediately above a; /??G, the tree II consisting of a; *
/3?G* and the subtree of II* with root r above it is mapped by *R* to an ici-tree. The tree II is represented by**

$$\frac{\textbf{Ep}\quad \textbf{SI}}{}$$

where we allow E_x to be empty in case r has only one premise. i2[E_0] and *R[*$\Sigma_1$are ici-trees by the induction hypothesis. Let E{, = #[E_0] and E'_x = *R*[Σ_1]. We proceed by case according to the rule r. Consider A t; here II is

$$\frac{\Sigma_0\left\{\begin{array}{c}\vdots\\ \alpha;\beta?\phi_1\end{array}\right.\qquad \Sigma_1\left\{\begin{array}{c}\vdots\\ \alpha;\beta?\phi_2\end{array}\right.}{\alpha;\beta?\phi_1\wedge\phi_2}$$

If *R[a; /3?<t>i]* = a'; /?'?# for each f = 1,2, then *R[a; 0?fa A<fo] = a';* £'?# A#, and *R[U]* is

$$\frac{\Sigma'_0\left\{\begin{array}{c}\vdots\\ \alpha';\beta'?\phi'_1\end{array}\right.\qquad \Sigma'_1\left\{\begin{array}{c}\vdots\\ \alpha';\beta'?\phi'_2\end{array}\right.}{\alpha';\beta'?\phi'_1\wedge\phi'_2}$$

this is, by A t> an ici-tree. All other propositional rules follow in a similar way. Now let us-consider the quantifier rules. In the case of V 4-, II is

$$\frac{\text{E}_\circ\left\{\begin{array}{c}\vdots\\ \text{1 } cr,p,<t>z?G\end{array}\right.}{\alpha;\beta?G}$$

where (Vx)^a; € *a/3*. Let *t* = *R[z]*. If *a[z]* € £1, then *t* = *a[z]* € A, by the definition of *R;* if not, then *t* = *x* for some appropriate *x* 6 A. In either case, t e £1 and i2[E_0] = Eg' is an ic_x-tree for a^7; /?, <£'t?G". -R[II] is then

$$\frac{\Sigma'_0\left\{\begin{array}{c}\vdots\\ \alpha';\beta',\phi't?G'\end{array}\right.}{\alpha';\beta'?G'}$$

where (Vx)^'x € *a'ft'*. By V 4-, *R[R]* is an ici-tree. The case for 3 t is similar. Finally, consider 3 J.; here II is

$$\frac{\Sigma_0\left\{\begin{array}{c}\vdots\\ \alpha,\phi f\overline{z};\beta?G\end{array}\right.}{a;0?G}$$

where *(3x)<j>x e a/3* and *f̄z $ Tsn(<*P,G)*. There is no *z e Tsn((*P,G)* such that *a[z]* = *a[fz̄]*, because any such *z* would be an argument of *f*.[22] Thus, ***R[fz̄] ? T$_S$H(a/3,G)*. Say *R[fz̄] = x; R[I1}* is now**

$$\dfrac{\Sigma'_0\left\{\begin{array}{c}\vdots\\ \alpha',\phi'x;\beta'?G'\end{array}\right.}{a']/3^{r}?G'}$$

R[U] is an ici-tree by 3 |, since *(3x)</>'x* G *a'/3^f*, and *x* is new to a', *G^1*. The case for V f is similar. E

Having associated with partial icsH-trees partial ici-trees, we show now that the "association" preserves global correctness in the sense of the following theorem:

Valuation Theorem. *Let* T *be an* icsH-*ree> ^{a} ^{a} *substitution.*

(i) *IfY$_a$e[Tl then[R$_a$[T]]=Y.*

(ii) // [T] = 0, *then [R*[T]]* = U *or* N.

PROOF. Let T"and *a* be given as above; we use *R* as an abbreviation for -Rj. To establish the theorem it suffices to show by induction on II that for every subtree II of T (rooted at a question node):

Y$_p$ G **[n]$_T$** for some *p* ≤ *a* if and only if [flplU^Tj = Y.

Note, that if *Y$_p$* G [IIJT for some *p* ≤ a, then /2[II] is an ici-tree by the local correctness lemma. Let *N* = a; /??G be the root of II. For the base case assume that *N* is a leaf node. If *Y$_p$* € **[n]$_T$** for some *p* ≤ *a*, then choose ***<f> € afi* such that p{0] = *p[G*. *a[<f>] = ap{<() = a[p[<t>]] = (?[p[G]] = <?p[G\ =*** *a[G]*. Thus *R[G\ e R[ap]* up to renaming of bound variables, so *[R[U]]* = Y. Conversely, assume *[R[U]]* = Y. Then *R[G]* G *R[a(3]* up to renaming of bound variables, so Y$_p$ G [a; *p?G*.
In the inductive step, *N* is not a leaf node. For the ^-direction, assume Y$_p$ G *[N]T* for some *p* ≤ *a*. Let *M* be any node immediately above *N* such that Y$_p$ G *[M*. (Such an *M* exists by the definition of'[JV].) If *M* has a

---

[22] Here one uses the fact that, given any term *f(z\,..., z$_n$)*, there is no substitution *a* such that *<r[f(zi,..., z$_n$)]* = *a[zi]* for any 1 ≤ t ≤ n.

single premise M_o, then Y_p G ***[M$_o$]*** by the definition of **[M].** By inductive hypothesis, *[R[M$_0$]]* = Y. Then *[R[M]]* = Y and hence *[R[N]]* = *Y* as well. If M has two premises, M_o and M_x, then by the definition of **[M],** there are po and p_x such that Y^ G **[M$_o$],** Y_{Pl} G **[MJ,** and *p* = poVpi- Since Po _< P _< <r, we have *po* _< *a* and, thus, [#[M_0]] = Y. Similarly we have [flfMx]] = Y. But then *[R[M]]* = *Y* and hence [/2[JV]] = Y.
For the ^-direction, assume [i?[iV]] = Y, and choose rule node M immediately above *N*, such that [#[M]j = Y. If M has only one premise M_o, then [i2[M_0]] = Y. By inductive hypothesis, Y_p G [M$_0$]T for some *p* _< *a*. [TV] = [M] = [Mo], so we are done. If M has two premises Mo and Mi, then *[R[M$_0$]]* = Y and *[R^]]* = Y by the definition of *[R[M]]*. By the inductive hypothesis, choose (for t = 0,1) *pi* such that Y_{Pi} G **[M$_t$]** and p_t _< a. ^/>ovpi € [A^J ^ **[^VJ-** Since *PQ<G* and pi _< a, poVpi _< a; so we are done.

D

The following corollary to this theorem establishes the usefulness of the ICSH calculus.

Correctness for ICSH- *Let* a, *G be in C\; then there is an icsn-derivation for a?G if and only if there is an ici-derivation for a?G.*

Indeed, the nd-proofs that are then associated with ici-derivations are, depending on the operation chosen in the ±-rules, either p-normal or normal. The SH-expansion is thus a tool that provides correctly nd-proofe. In the introductory remarks to the previous section we mentioned that the SH-expansion is to be used for proof search, indeed, proof search that extends in a most natural way the strategic considerations for sentential logic—implemented in the Carnegie Mellon Proof Tutor. Those strategic considerations are described in [Sieg and Schemes]; here we review just the coarse structure of the (very efficient) search procedure. The search for an answer, i.e., an ic-derivation, to the question *a;/3?G* involves three distinct components: (i) use of i-rules, (ii) use of t-rules, (iii) use of _L-rules (with a limited set of contradictory pairs of formulas). It is step (i) that is central and taken in a *goal-directed way.* If the question

(*) Is *G* a strictly positive subformula of a formula in *a/3?*

has an affirmative answer, this step provides sequences of 4,-rule applications that *extract G* from strictly positive occurrences of *G* in elements of *aft.* The connecting formula sequences consist of the major premises of the ^-r**ules** and require, in general, answers to new questions, namely, those raised in the minor premises of the rule applications.

It is for the appropriate generalization of this *extraction strategy* that the SH-expansion is absolutely critical. Recall that the question in sentential logic "Is *G* an element of a/??" is generalized in predicate logic to "Is *G* unifiable with an element of *aft?"*. The goal-directedness of applications of 4-rules is now obtained by generalizing the question (*) above to

> Is *G* unifiable with a strictly positive canonical subformula of a formula in ***a/3?***

A subformula is considered to be a *canonical* one, if quantifiers are instantiated by terms that match the ^-quantifier rules of ICSH> ie., those terms would be used, if the formula were "extractable" by 4-rules.—This natural extension of the sentential logical search satisfies three important desiderata: (i) logical truths of sentential logic, e.g., instances of the law of excluded middle with complex formulas of *d*, are recognized without appealing to quantificational rules; (ii) the selection of terms for V *I* and 3 t is delayed; (iii) extractability is the central feature of the search. The details of our approach to automated proof search will be presented in a later publication (together with a discussion of benchmark examples).

7. 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 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, we are convinced, fruitfully with a sophisticated, automated search for humanly intelligible proofs.

Appendix

In this appendix we give first a definition used at the end of section 3; then two diagrams are drawn that complement the text of sections 2 and 3.

***Positive* and *strictly positive subformulas* of a given formula are defined by induction; indeed, for the first concept one defines simultaneously, when ^ is a *positive subformula* of *rj}[<f>* G P(V0] or 0 is a *negative subformula* of $\psi[\phi$ G *N(ij;)]*, namely by the rules,**

(i) 0 is ^'=> ^ G *P{w)*

(") (a)</»is -«V»i, ^ G /yT(V»i) ==• ^ G P'(ψ)

(b)%> is -* , ^ / '(^) =• $ G ^'(ψ)

(iii) (a)t/»is V'l A V^t 0 G P(V'i) U P(t/»2) $\Longrightarrow \phi \in P(\psi)$

(b)i/»is V)j A^>2, <f> G N(V'i) U A/(V*2) =>-0G $N(\psi)$

(iv) (a)i*j)* is V'1 $\vee \psi_2, \phi$ GP(Vi)UP(^:)=^^G $P(\psi)$

0>)i/; is V'1 $\vee \psi_2, \phi$ GA/(^I)UA/(V2) $\Longrightarrow \phi \in N(\psi)$

(v) (a),/> is ^!)j —>• ^>2, <j^ G P(V^) = ^ ^G P(V)

(b)i /> is ^^1 ~^ V*2i <6 G iV(^i) =^ ^»GP(V>)

(c)i /; is V'1 $\rightarrow \psi_2$, ç^ G *P(ipi)* U A/(^^2) =^=>- 0 (E $N(\psi)$

Finally, ^ is a *strictly positive subformula* of V*[£ € «S(V>)] 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 illustrates the construction of canonical refutation branches discussed in section 3.

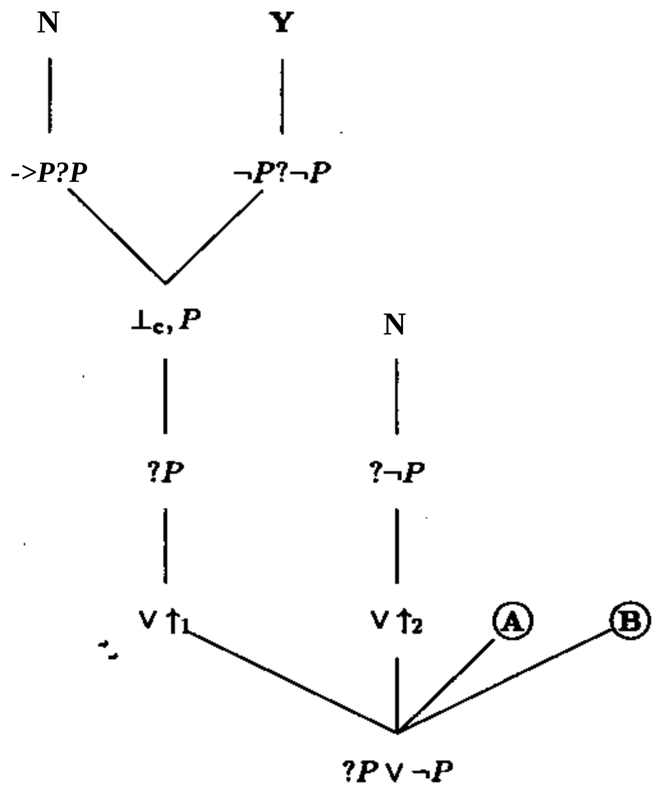
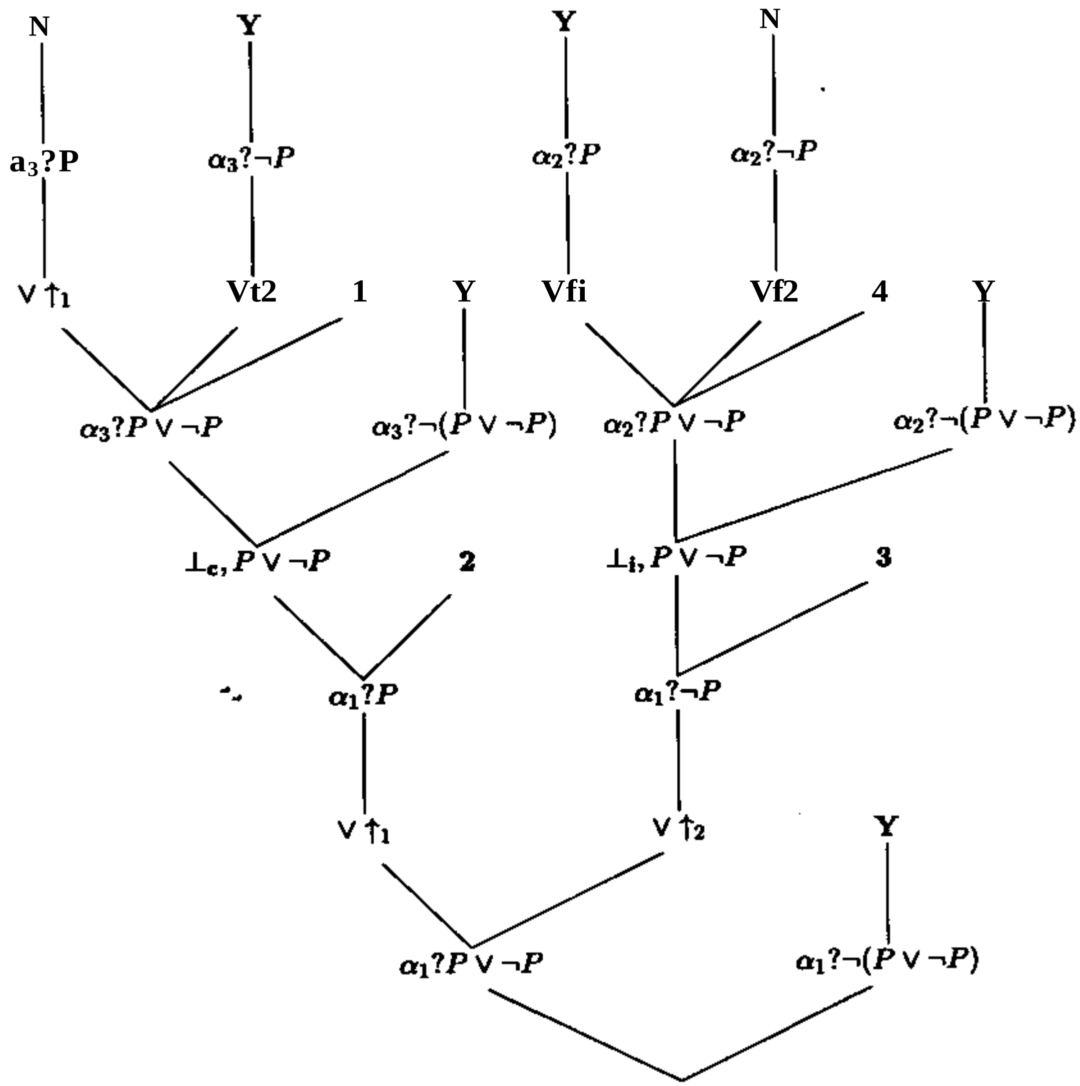


Diagram 1.

Diagrams 1.A and l.B expand nodes A and B, respectively.



$\alpha_1 = \neg(P \vee \neg P)$

$\alpha_2 = \neg(P \vee \neg P), P$

$\alpha_3 = \neg(P \vee \neg P), \neg P$

Diagram 1.A.

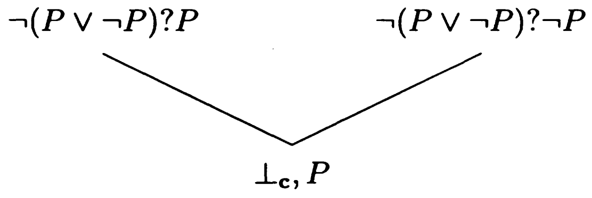


Diagram 1.B.

The question nodes above are expanded in exactly the same way as the corresponding questions in diagram 1.A.

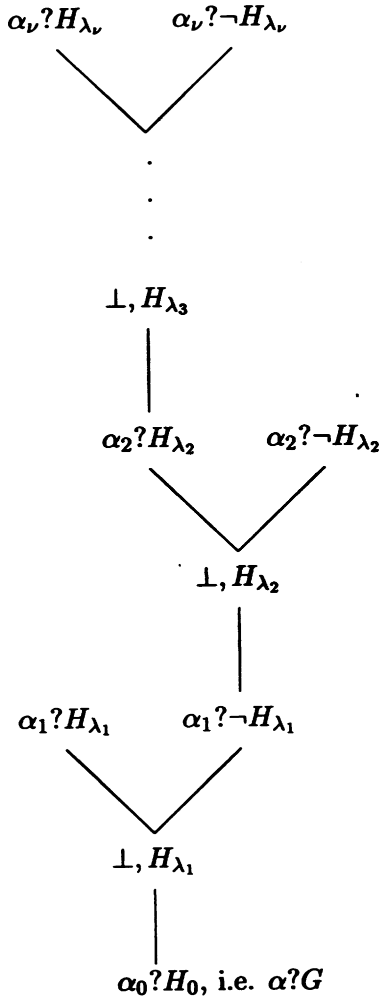

$\alpha_\nu ? H_{\lambda_\nu}$
$\alpha_\nu ? \neg H_{\lambda_\nu}$
$\perp, H_{\lambda_3}$
$\alpha_2 ? H_{\lambda_2}$
$\alpha_2 ? \neg H_{\lambda_2}$
$\perp, H_{\lambda_2}$
$\alpha_1 ? H_{\lambda_1}$
$\alpha_1 ? \neg H_{\lambda_1}$
$\perp, H_{\lambda_1}$
$\alpha_0 ? H_0$, i.e. $\alpha ? G$

Diagram 2.

References

[1] P. Andrews. Transforming matings into natural deduction proofs. In *5th Conference on Automated Deduction,* pages 281-292, New York, Berlin, 1980. Springer-Verlag.

[2] M. Baaz and C. G. Fermiiller. Non-elementary speedups between different versions of tableaux. In Baumgartner et al., editors, *Theorem Proving with Analytic Tableaux and Related Methods,* pages 217-230. Springer, 1995.

[3] W. Bledsoe. Non-resolution theorem proving. *Artificial Intelligence,* 9:1-35, 1977.

[4] W. Bledsoe. The UT natural-deduction prover. Technical Report, Departments of Mathematics and Computer Science, University of Texas, April 1983.

[5] S. Cittadini. Intercalation calculus for intuitionistic propositional logic. Carnegie Mellon Technical Report PHIL-29, Philosophy, Methodology, and Logic, 1992.

[6] R. Constable et al. *Implementing Mathematics with the Nuprl Proof Development System.* Prentice-Hall, Englewood Clifls, NJ, 1986.

[7] F. B. Fitch. *Symbolic Logic: An Introduction.* Ronald, 1952.

[8] M. Fitting. *First-Order Logic and Automated Theorem Proving.* Springer-Verlag, New York, Berlin, 1990.

[9] G. Gentzen. Untersuchungen fiber das logische Schliefien i,ii. *Math. Zeitschrift,* 39:176-210, 405-431,1934,1935. English translation in [11].

[10] G. Gentzen. Die Widerspruchsfreiheit der reinen Zahlentheorie. *Mathematische Annalen,* 112:493-565, 1936. English translation in [11].

[11] G. Gentzen. *The Collected Papers of Gerhard Gentzen.* North-Holland Publishing Company, Amsterdam, 1969. Edited by M. E. Szabo.

[12] J. Herbrand. *Logical Writings.* Cambridge, Harvard University, 1971. Edited by W. Goldfarb.

[13] S. Jaśkowski. On the rules of suppositions in formal logic. *Studia Logica,* (1), 1934.

[14] S.C. Kleene. *Introduction to Metamathematics.* Wolters-Noordhoff Publishing, Groningen, 1952.

[15] S. MacLane. *Abgekiirzte Beweise im Logikkalkiil* PhD thesis, University of Göttingen, 1934.

[16] S. MacLane. A logical analysis of mathematical structure. *The Monist,* pages 118-130,1935. The paper was read to the American Mathematical Society on December 28, 1933.

[17] A.J. Nevins. A human oriented logic for automatic theorem proving. J. *ACM,* 21:606-621, 1974.

[18] L. Paulson. *Logic and Computation: Interactive Proof with Cambridge LCF:* Cambridge University Press, 1987.

[19] J. Pelletier. Automated natural deduction in Thinker. This volume.

[20] F. Pfenning. *Proof Transformations in Higher-Order Logic.* PhD thesis, 1987. Carnegie Mellon University.

[21] F. Portoraro. Strategic construction of Fitch-style proofs. This volume.

[22] D. Prawitz. *Natural Deduction: A Proof-Theoretical Study.* Almqvist & Wiskell, Stockholm, 1965.

[23] N.A. Shanin et al. An algorithm for a machine search of a natural logical deduction in a prepositional calculus. In Siekmann and Wrightson, editors, *Automation of Reasoning, vol.* i, pages 424-483. Springer-Verlag, New York, Berlin, 1983. Reprinted from Izdat. Nauka, Moscow, 1965.

[24] W. Sieg. *Mechanisms and Search: Aspects of Proof Theory.* Associazione Italiana di Logica e sue Applicazioni, Padova, 1992.

[25] W. Sieg. Intercalation calculi for classical logic. Carnegie Mellon Technical Report PHIL-46, Philosophy, Methodology, and Logic, 1994.

[26] W. Sieg and B. Kauffmann. Unification for quantified formulae. Carnegie Mellon Technical Report PHIL-44, Philosophy, Methodology, and Logic, 1993.

[27] W. Sieg and R. Scheines. Searching for proofs (in sentential logic). In L. Burkholder, editor, *Philosophy and the Computer,* pages 137-159. Westview Press, Boulder, San Francisco, Oxford, 1992.

[28] W. Snyder. *A Proof Theory for General Unification,* Birkhäuser, Boston, Basel, Berlin, 1991.

[29] G. Stålmarck. Normalization theorems for full first order classical natural deduction. *J. Symbolic Logic,* 56:129-149, 1991.

[30] R. Statman. *Structural Complexity of Proofs.* PhD thesis, Stanford, 1974.

[31] A. S. Troelstra and D. van Dalen. *Constructivism in Mathematics: An Introduction.* North Holland, Amsterdam, 1988.