ON LINEAR NATURAL DEDUCTION

Daniel Leivant

TR 79-400

Department of Computer Science
Cornell University
Ithaca, New York 14853

ON LINEAR NATURAL DEDUCTION

Daniel Leivant *

Computer Science Department
Cornell Univeristy
Ithaca, N.Y. 14853

Since Gentzen introduced natural deduction (ND) in 1934 this type of logical calculus has enjoyed recurring surges of interest, for pedagogical reasons if not for technical or philosophical ones. The repudiation of Hilbert-type calculi for tutorial aims has stemmed from the nature of the axioms used, these being set up to maximally embody the combinatorial properties of the logical operations (leading eventually to their explicit isolation in Combinatory Logic). In contradistinction, the rules of ND are, in a sense, minimal (cf. [Pr 65]). The combinatory complexity of the logical machinary is embodied in the structure of proofs (prfs) as a whole, rather than in axioms. Also, ND clearly separates logical rudiments, given by inference rules, from mathematical rudiments, given by axioms.

Originally, Gentzen presented natural deductions as trees, a form that permits an explicit display of the flow of argument. While increasing the "naturalness" in one respect, the tree format is divorced from the usual style of informal argumentation, which is linear. Many have later opted for the reverse, trading off direct

* The author is grateful to Joseph Bates and to Robert Constable for useful comments on a draft of this note.

flow of argument for linearity.

The appropriateness of linear prfs for display in a linear context (e.g., book, computer print out) is obvious. As a by-product, verification of syntactic constraints is often easier. In addition, linearity sometimes provides economy, as for certain deductions that have the tree form

$$\dfrac{\begin{array}{cc}\Pi & \Pi \\ A & A \\ \Pi_1 & \Pi_2 \\ B & C\end{array}}{B \wedge C}$$

A linear verison of this might present $\underset{A}{\Pi}$ only once.

The main difficulty with linearizing ND lies in the necessity to derive a formula (fml) from premises occurring at any distance back in the prf. In a Hilbert-type system this creates no difficulty since the proof has no "global structure". In contrast, in ND a fml F may not be arbitrarily derived from previous occurring premises, since, e.g., these may depend on assumptions closed before F, causing a "change in the logical environment", so to speak.

Another viewpoint on the difficulty is this. In the tree formalism, horizontal co-occurrence of fmls is interpreted as (metamathematical) conjunction, and vertical ordering--as implication (Disjunction has no structural representation, a seed of trouble.)

Once deductions are compressed into a linear form, a single dimension is used to interpret, ambiguously, both conjunction and implication.

In what follows we discuss three approaches for dealing with the difficulties of linearizing ND. The first is based on a reference to sequents $\Gamma \Rightarrow F$ (where Γ is a set of fmls), in place of single fmls (§1). The second is a widely used block formalism with accessibility conditions on inferences; we present an inductive definition of such calculi, that does not use accessibility conditions (§2). Finally, we present a formalism using fmls (not sequents) in which global conditions on prf correctness are minimized, and separated from other conditions (§ 3). We compare the three styles, and maintain that the sequential variant is superior to the others (§ 5). We conclude by suggesting a criterion for inserting derived rules into ND calculi (§ 6), and by discussing alternatives to ND (§ 7).

This introduction would not be complete without a word of caution. Although linearizing ND is useful for implementation, it is defective in other respects. Gentzen's original tree format remains conceptually clearer and more coherent than any linear variant, and therefore pedagogically superior. Furthermore, linear ND is practically useless in deriving metamathematical results. The starting point for such applications is the Normal Form Theorem (cf. [Pr 65]), from which one derives, e.g., interpolation, intutionistic independence of logical operators, disjunction and existential instantiation properties, and Herbrand's Theorem. Although the usual normalization techniques may be adapted to linear variants by

brute force, they lose too much of their clarity to remain an acceptable alternative.

1. A sequential approach.

A natural deduction Δ does not prove, in general, its last fml F, but derives F from certain assumptions $A_1,\ldots,A_k$. Thus, Δ derives a sequent $\{A_1,\ldots,A_k\} \Rightarrow F$. The sequent is the local embodiment of the global character of the derivation. By using sequents in place of fmls as the basic entities, the global features of ND are totally localized. For instance, the rules for $\rightarrow$I, $\vee$E, $\forall$I and $\exists$E read as follows.

$\rightarrow$I : $\dfrac{\Gamma, A \Rightarrow B}{\Gamma \Rightarrow A \rightarrow B}$ (here Γ, A abbreviates $\Gamma \cup \{A\}$).

$\vee$E : $\dfrac{\Gamma_0 \Rightarrow A_1 \vee A_2 \quad \Gamma_1, A_1 \Rightarrow B \quad \Gamma_2, A_2 \Rightarrow B}{\Gamma \Rightarrow B}$ (where $\Gamma = \Gamma_0 \cup \Gamma_1 \cup \Gamma_2$)

$\forall$I : $\dfrac{\Gamma \Rightarrow A}{\Gamma \Rightarrow \forall_x A}$ provided x does not occur in Γ.

$\exists$E : $\dfrac{\Gamma_0 \Rightarrow \exists x A \quad \Gamma_1, A \Rightarrow B}{\Gamma \Rightarrow B}$ provided x does not occur in Γ,

The linearization of the resulting calculus is trivial, since there are no global conditions to worry about. Note that the calcul is not a " sequent calculus" in the sense of Gentzen, since we do not have rules operating on antecedents, only on succedents of sequents (cf. § 7 below).

As an example, we give an intuitionistic proof of $\neg\neg(A \vee \neg A)$, where $\neg F$ is an abbreviation for $F \rightarrow \bot$.

1.	$\neg(A \vee \neg A)$	$\Rightarrow$	$\neg(A \vee \neg A)$	
2.	A	$\Rightarrow$	A	
3.	A	$\Rightarrow$	$A \vee \neg A$	2, $\vee$I
4.	$\neg(A \vee \neg A), A$	$\Rightarrow$	$\bot$	1,3, $\rightarrow$E
5.	$\neg(A \vee \neg A)$	$\Rightarrow$	A	4, $\rightarrow$I
6.	$\neg(A \vee \neg A)$	$\Rightarrow$	$A \vee \neg A$	5, $\vee$I
7.	$\neg(A \vee \neg A)$	$\Rightarrow$	$\bot$	1,6, $\rightarrow$E
8.		$\Rightarrow$	$\neg\neg(A \vee \neg A)$	7, $\rightarrow$I

The use of this system in practice is impeded by the necessity to re-write assumptions *ad nausea* in longer proofs. One may often drop repetitions, noting only the important changes in sequents' antecendents; but this is not very feasable when remote premises are involved. However, what seems to be a difficulty turns out to be an asset in a computer environment. The user may write the proof with pointers to the premises, leaving it to the machine to search back for the assumptions (a trivial algorithm!), and display them in a compiled version of the proof. The proof's writer may mention as few or as many assumptions as he wishes; even the bare, concise list of succedent-fmls would be accepted by the machine as correct syntax, and verified to be a valid derivation.

Moreover, the machine may provide additional editing facilities which would make it easier for the proof's reader to check details, or skip details and read the uninterrupted core of the proof. For

instance, assumptions may be listed by their reference numbers in the proof; only assumptions active in deriving a sequent might be displayed separately (this may be omitted if the assumption is displayed within the x preceding lines). The text may be indented with each new assumption, with out-denting when an assumption is closed; assumptions whose readibility is impeded by in- and out-denting may be repeated, etc., etc.

For the bare core of the proof above, a possible output would be the following edited text.

inactive assumptions	active assumptions		core	premises	remote
	ass	1.	$\neg(A \vee \neg A)$		
	ass	2.	A		
2		3.	$A \vee \neg A$	2	
1	2= A	4.	$\bot$	1,2	
1		5.	$\neg A$	4	
1		6.	$A \vee \neg A$	5	
	1= $\neg(A \vee \neg A)$	7.	$\bot$	1,6	1= $\neg(A \vee \neg A)$
		8.	$\neg\neg(A \vee \neg A)$	7	

The use of elaborate source analysis is, for short derivations, more of a nuisance than a help. But there is any reason to believe that it would be of great help for composing and reading mammoth proofs. Again, the calculus is flexible enough to allow an easy switch from one type of implementation to another.

2. The block and accessibility conditions approach.

The most common approach to linearization of ND is the delineation of sub-proofs ("blocks"), accompanied by restrictions on the availibility (="accessibility") of premises at a given point in the prf. Different authors delineate blocks by various devices. Philosophy textbooks usually use a vertical line along the left magin of a block, or simply a rectangular box surrounding it. Such typographic eccentricities are a nuisance, especially for computer implementation. The use of verbal parentheses in [CO], borrowed from the syntax of programming languages, is certainly more appropriate.

An aspect of these calculi that may seem surprising is the insistance on a Hilbert-type definition, in the sense that every fml (other than assumptions and mathematical axioms) is claimed to be derived from previously occurring fmls by one of the inference rules. With this as the underlying mechanism, restrictions of a global nature are imposed on the prf. Unfortunately, the project's outcome does not conform with the stated intention. Some inference rules use fmls as premises, but some other use a prf ($\rightarrow$I) or a combination of fmls and prfs (vE). Moreover, the actual act of infering involves, besides writing down the conclusion, also a syntactic delineation of the prf-block(s) used as premise(s).

This inconsistency is not without its reasons. Clearly, what one would like to have here is an inductive definition of the notion "proof." An accurate definition of Gentzen's de-

ductions should also be inductive; e.g., if $\begin{matrix}A\\ \Pi\\ B\end{matrix}$ is a prf,

then so is $\dfrac{\begin{matrix}(n)\\ A\\ \Pi\\ B\end{matrix}}{A \quad B}(n)$ (cf. [Pr 65]). More than a mere pedant's technicality, this may be viewed as a constructivist explication of the logical operations (cf. [Pr 71] § II.2). Unfortunately, a similar inductive definition for the linear version is defective in two respects. Firstly, as for the tree-version, several occurrences of the same assumption may be closed (in derivations so defined) by a single instance of, say, $\rightarrow$I, making the parsing of a derivation unpleasant. Secondly, the potential economy of the linear style is dashed: each use of a fml F as premise necessitates a separate display of a (sub-) prf of F.

It seems worthwhile to present a simple inductive definition of linear ND calculi, of the forementioned type, that does permit the desired economy. Let Γ, Δ denote (finite) sets of fmls; we write $\Gamma \Rightarrow \Delta$ to express that each $D \in \Delta$ is derived from the assumptions Γ (NB: this convention disagrees with the usual interpretation of sequents). Now define linear ND by the following inductive clauses.

1. $\{A\}$ is a prf for $\{A\} \Rightarrow \{A\}$.
2. ($\wedge$I). If Π_i is a prf for $\Gamma_i \Rightarrow \Delta_i$, $A_i \in \Delta_i$, $i = 1,2$,

then $\begin{matrix}\Pi_1\\ \Pi_2\\ A_1 \wedge A_2\end{matrix}$ and $\begin{matrix}\Pi_2\\ \Pi_1\\ A_1 \wedge A_2\end{matrix}$ are prfs for $\Gamma \Rightarrow \Delta \cup \{A_1 \wedge A_2\}$,

where $\Gamma = \Gamma_1 \cup \Gamma_2, \Delta = \Delta_1 \cup \Delta_2$.

3. Similar clauses for ($\wedge$E), ($\rightarrow$E), ($\vee$I), ($\bot$), ($\bot_c$), ($\forall$E), ($\exists$I).

4. ($\rightarrow$I) If Π is a prf for Γ, $A \Rightarrow \Delta$, $B \in \Delta$, then $\underset{A \rightarrow B}{\Pi}$ is a prf for $\Gamma \Rightarrow \{A \rightarrow B\}$. (Note: fmls in Δ are not any longer available for subsequent construction.)

5. ($\vee$E) If Π_0 is a prf for $\Gamma_0 \Rightarrow \Delta_0$, $A_1 \vee A_2 \in \Delta_0$, and Π_i is a prf for Γ_i, $A_i \Rightarrow \Delta_i$, $B \in \Delta_i$, $i = 1,2$, then $\begin{matrix}\Pi_0\\ \Pi_1\\ \Pi_2\\ B\end{matrix}$ and $\begin{matrix}\Pi_0\\ \Pi_2\\ \Pi_1\\ B\end{matrix}$

are prfs for $\Gamma \Rightarrow \Delta_0, B$, where $\Gamma = \Gamma_0 \cup \Gamma_1 \cup \Gamma_2$.

<u>Stronger variant</u>: If Π_0 is a prf for $\Gamma_0 \Rightarrow \Delta_0$, $A_1 \vee A_2 \in \Delta_0$, and $\begin{matrix}\Pi_0\\ \Pi_i\end{matrix}$ is a prf for Γ_i, $A_i \Rightarrow \Delta_i$, $B \in \Delta_i$, $i = 1,2$, then $\begin{matrix}\Pi_0\\ \Pi_1\\ \Pi_2\\ B\end{matrix}$

and $\begin{matrix}\Pi_0\\ \Pi_2\\ \Pi_1\\ B\end{matrix}$ are prfs for $\Gamma \Rightarrow \Delta_0, B$, where $\Gamma = \Gamma_0 \cup \Gamma_1 \cup \Gamma_2$.

6. ($\forall$I) If Π is a prf for $\Gamma \Rightarrow \Delta$, $A \in \Delta$, x does not occur free in Γ, then $\underset{\forall xA}{\Pi}$ is a prf for $\Gamma \Rightarrow \{\forall xA\}$.

<u>Stronger variant</u>: Let $[x]\Delta := \{F \in \Delta \mid x \text{ not free in } \Gamma\}$. If Π is a prf for $\Gamma \Rightarrow \Delta$, $A \in \Delta$, x not free in Γ, then $\underset{\forall xA}{\Pi}$ is a prf for $\Gamma \Rightarrow [x]\Delta, \forall xA$.

7. ($\exists$E) If Π_0 is a prf for $\Gamma_0 \Rightarrow \Delta_0$, $\exists xA \in \Delta_0$, and Π_1 is a prf for Γ_1, $A \Rightarrow \Delta_1$, $B \in \Delta_1$, x not free in Γ_1, then $\begin{matrix}\Pi_0\\ \Pi_1\\ [B]\end{matrix}$ is a prf for $\Gamma_0 \cup \Gamma_1 \Rightarrow \Delta_0, B$. ([B] means: B is optional).

Stronger variant: If $\begin{matrix}\Pi_0\\ \Pi_1\end{matrix}$ is a prf for $\Gamma_1, A \Rightarrow \Delta_1$, and Π_0 is a prf for $\Gamma_0 \Rightarrow \Delta_0$, $\exists xA \in \Delta_0$, x not free in Γ_1, then $\begin{matrix}\Pi_0\\ \Pi_1\end{matrix}$ is a prf for $\Gamma_0 \cup \Gamma_1 \Rightarrow \Delta_0, [x]\Delta_1$.

Classical variant: If Π is a prf for $\Gamma \Rightarrow \Delta$, $\exists xA \in \Delta$, z not free in Π, then $\begin{matrix}\Pi\\ A[z/x]\end{matrix}$ is a prf for $\Gamma \Rightarrow \Delta, A[z/x]$. (this rule necessitates a restriction on $\forall$I (6): the proper variable x there may not be the proper variable of any $\exists$E in Π)

Examples: 1.

$\forall x.\ Ax \wedge C$
$Ax \wedge C$
C
Ax
$\forall x Ax$
$\forall x Ax \wedge C$
$\forall x.Ax \wedge C \rightarrow \forall x Ax. \wedge C$

This prf uses the stronger variant of $\forall$I. A prf using the weaker variant:

$\forall x.Ax \wedge C$
$Ax \wedge C$
Ax
$\forall x Ax$
$\forall x.Ax \wedge C$
$Ax \wedge C$
C
$\forall x Ax \wedge C$
$\forall x.Ax \wedge C \rightarrow \forall x Ax. \wedge C$

2.

$A \rightarrow C. \wedge . B \rightarrow C$
$A \vee B$
A
$A \rightarrow C$
C
B
$B \rightarrow C$
C
C
$A \vee B \rightarrow C$
$A \rightarrow C. \wedge . B \rightarrow C. . \rightarrow . A \vee B \rightarrow C$

This prf uses the stronger variant of $\vee E$. A prf using the weaker variant:

$$
\begin{array}{c}
A \vee B \\
A \\
A \to C . \wedge . B \to C \\
A \to C \\
C \\
B \\
A \to C . \wedge . B \to C \\
B \to C \\
C \\
C \\
A \vee B . \to C \\
A \to C . \wedge . B \to C . . \to . A \vee B \to C
\end{array}
$$

Let us now consider the parsing of the system. This may be done by successively delineating subproofs for each use of the inductive clauses. Such a complete parsing is quite cumbersome, as the nesting of proof-blocks grows as fast as the height of the corresponding tree-format ND. Instead, one may single out the restrictive clauses $\to I$, $\vee E$, $\forall I$, $\exists E$; these are restrictive in the sense that the prf generated proves a sequent with a succedent generally weaker than the union of the succedents corresponding to the input prfs. Using verbal parentheses just for the prfs generated by restrictive clauses, one is able to recognize the syntactic correctness of a prf by "accessibility conditions" (as in [CO]), rather than by inspection of the inductive definition.

3. Making the most of linearity with local conditions.

Here we attempt to make the most out of linear ND defined via local conditions. Firstly, we give a systematically local defini-

tion of prfs, in the sense that the premises of a line in the prf are always previous lines, not blocks with global structure. We isolate the indispensible global conditions, using them solely to define the closure of assumptions. Secondly, we reduce structural restrictions, i.e.: we allow more prfs to be identified as correct.

A prerequisite for understanding the resulting calculus is familiarity with the intended global structure, so the appeal of the system is in its simple implementation and flexibility, not in pedagogical qualities.

Let A,B etc. stand for fmls, x,y for variables, t--for a term. An assumption is an expression of the form "ass A". A marker is an expression of the form "close A" or "for x". A pseudo-prf (pprf) is a list of fmls (including assumptions) and markers, to which we refer jointly as lines. A line in a pprf is accessible from a subsequent line if any intermediary line "close A" is preceded by an intermediary line "ass A" (for any fml A). The inference rules are the following.

$\wedge$I: $\dfrac{A_0 \quad A_1}{A_0 \wedge A_1}$; $\wedge E_i$: $\dfrac{A_0 \wedge A_1}{A_i}$ $\quad i = 0,1.$

$\rightarrow$I: $\dfrac{B}{[\text{close } A,]\ A \rightarrow B}$; $\rightarrow$E: $\dfrac{A \rightarrow B \quad A}{B}$

$\vee$I: $\dfrac{A_i}{A_0 \vee A_1}$, $i = 0,1$; $\vee E_1$: $\dfrac{A_0 \vee A_1}{\text{close } A_i,\ \text{ass } A_j}$ $(i,j) = (0$ or $(1$

; $\vee E_2$: $\dfrac{G,\ \text{close } A_0,\ \text{ass } A_1 \quad G}{[\text{close } A_1,]\ G}$

$\forall I$: $\dfrac{A(z)}{\forall x A(x),\ \text{for } z}$; $\forall E$: $\dfrac{\forall x A(x)}{A(t)}$

$\exists I$: $\dfrac{A(t)}{\exists x A(x)}$; $\exists E_1$: $\dfrac{\exists x A x}{\text{ass } A z,\ \text{for } z}$

$\exists E_2$: $\dfrac{\text{ass } A(z),\ \text{for } z \quad G}{[\text{close } A(z),] G}$

For the intuitionistic system:

$\bot$: $\dfrac{\bot}{A}$

For the classical system:

$\bot$; $\dfrac{\bot}{[\text{close} \neg A,] A}$

A prf is a pprf where each line other than assumptions is derived from previous accessible lines by one of the inference rules, in the following sense. When premises of an inference rule are displayed side by side, they may occur in any order; premises that are ordered (vertically) should occur in that order; premises or conclusions that are separated by commas should be adjacent in the prf; expressions appearing in brackets are optional.

A block (for A) in a pprf Π is a sub-pprf of Π with "ass A" as the first line, and "close A" as the last line. We define by induction on the length of a block β for A what it means for β to be closed. β is closed if each marker "ass B" in β is the first line of a closed sub-block of β, and if no variable free in A is

marked in β . An assumption is <u>closed</u> if it is the first line of a closed block.

It is simple to see that every theorem of first-order intuitionistic (respectively, classical) logic is derived in the calculus described above by an intuitionistic (classical) prf in which all assumptions are closed. The small number of syntactic restrictions used in the definition makes the converse less obvious. In the following theorem we prove that the calculus is sound, and also give an interpretation to prfs in which not all assumptions are closed.

For a prf Π write

Γ_Π - for the assumptions not closed in Π

Δ_Π - for the lines in Π accessible in Π, i.e. - at the root of Π.

For $F \in \Gamma_\Pi$, let $F^\forall_\Pi$ be the closure of F for variables marked under "ass F" in Π.

Finally, $\Gamma^\forall_\Pi := \{F^\forall_\Pi \mid F \in \Gamma_\Pi\}$.

THEOREM: Let Π be a prf in the calculus above. Then

(1) $\Gamma^\forall_\Pi \vdash D$ for each $D \in \Delta_\Pi$.

The provability is in intuitionistic (classical) first-order logic if Π is intuitionistic (classical, respectively).

PROOF: We use induction on the length of Π, considering cases for the rule by which the root of Π is inferred.

1. The root is an assumption, $\Pi = \underset{\text{ass } A}{\Pi_0}$. Then $\Gamma_\Pi = (\Gamma_{\Pi_0}, A)$, $\Gamma^\forall_\Pi = (\Gamma^\forall_{\Pi_0}, A)$ $\Delta_\Pi = (\Delta_{\Pi_0}, A)$ and (1) is immediate by induction assumption for Π_0 (which may be vacuous).

2. <u>$\wedge$I.</u> $\Pi = \underset{A_1 \wedge A_2}{\Pi_0}$. Then $\Gamma_\Pi = \Gamma_{\Pi_0}$, $\Gamma^\forall_\Pi = \Gamma^\forall_{\Pi_0}$, $\Delta_\Pi = (\Delta_{\Pi_0}, A_1 \wedge A_2), A_1, A_2 \notin \Delta_{\Pi_0}$.

By ind.ass. $\Gamma^\forall_{\Pi_0} \vdash D$ for $D \in \Delta_{\Pi_0}$. Since $A_1, A_2 \notin \Delta_{\Pi_0}$, we also get $\Gamma^\forall_\Pi \vdash A_1 \wedge A_2$.

3. The cases for $\wedge$E, $\rightarrow$E, $\vee$I, $\bot$, $\forall$E and $\exists$I are similar.

4. <u>$\rightarrow$I.</u> $\Pi = \underset{\substack{\text{close } B \\ B \rightarrow C}}{\Pi_0}$

Case 4.1. The marker close B closes a block for B, with no initial closed sub-block.

$$\Pi = \Pi_0 \left\{ \begin{array}{l} \Pi_1 \\ \text{ass } B \\ \Pi_2 \end{array} \right. \begin{array}{l} \\ \\ \text{close } B \\ B \rightarrow C \end{array}$$

where Π_2 is the largest such block.

Then $\Gamma_{\Pi_0} = (\Gamma_{\Pi_1}, \Gamma_{\Pi_2}) = (\Gamma_\Pi, B)$; $\Delta_\Pi = (\Delta_{\Pi_1}, B \rightarrow C)$, and by the case's condition on the block, $\Gamma_{\Pi_1} \subset \Gamma_\Pi$. By ind.ass.

$\Gamma^{\forall}_{\Pi_1} \vdash D$ for each $D \in \Delta_{\Pi_1}$. For $F \in \Gamma_{\Pi_1}$, $F^{\forall}_{\Pi}$ may differ from $F^{\forall}_{\Pi_1}$ only in the presence of additional universal quantifiers; hence $\Gamma^{\forall}_{\Pi} \vdash D$ for each $D \in \Delta_{\Pi}$, $D \not\equiv B \to C$. Also, by ind.ass. applied to Π_0, $\Gamma^{\forall}_{\Pi_0} \vdash G$.

Since the block is closed, $B^{\forall}_{\Pi_0} = B$; so $\Gamma^{\forall}_{\Pi_0} = (\Gamma^{\forall}_{\Pi}, B)$, and $\Gamma^{\forall}_{\Pi} \vdash B \to C$.

Case 4.2. Otherwise; then $\Gamma_{\Pi} = \Gamma_{\Pi_0}$, $\Gamma^{\forall}_{\Pi} = \Gamma^{\forall}_{\Pi_0}$, $\Delta_{\Pi} \subseteq (\Delta_{\Pi_0}, B \to C)$, $C \in \Delta_{\Pi_0}$

By ind.ass. $\Gamma^{\forall}_{\Pi} \vdash D$ for $D \in \Delta_{\Pi_0}$, and with $D \equiv C$ we get also $\Gamma^{\forall}_{\Pi} \vdash B \to C$.

5. $\forall E_1$.

$$\Pi = \begin{array}{l} \Pi_1 \\ \text{ass } A_1 \\ \Pi_2 \\ \text{close } A_1, \text{ ass } A_2 \end{array}$$

Case 5.1. The marker "close A" closes a block Π_2 (with no initial closed sub-block) is the largest one corresponding to the inference. Then $\Delta_{\Pi} = (\Delta_{\Pi_1}, A_2)$, and by the case's condition, $\Gamma_{\Pi_1} \subseteq \Gamma_{\Pi}$. By ind.ass. $\Gamma_{\Pi_1} \vdash D$ for each $D \in \Delta$

For each $F \in \Gamma_{\Pi_1}$ $F^{\forall}_{\Pi}$ differs from $F^{\forall}_{\Pi_1}$ only in additiona universal quantifiers. So $\Gamma^{\forall}_{\Pi} \vdash D$ for each $D \in \Delta_{\Pi_1}$, while for $D \equiv A_2$, $A_2 \in \Gamma_{\Pi}$, so $\Gamma^{\forall}_{\Pi} \vdash A_2$ trivially.

Case 5.2. Otherwise; then $\Gamma_\Pi = (\Gamma_{\Pi_0}, A_2)$, $\Delta_\Pi = (\Delta_{\Pi_0}, A_2)$ and (1) is trivial by ind.ass.

6. $\vee E_2$.

$$\Pi = \begin{array}{l} \Pi_0 \\ \\ \\ \\ \\ \end{array} \begin{array}{l} \Pi_1 \\ \text{close } A_1,\ \text{ass } A_2 \\ \Pi_2 \\ \text{close } A_2,\ C \end{array}$$

Case 6.1. The marker "close A_2" closes a block, which, w.l.o.g., is the one corresponding to the inference (for else some lower occurrence α of "ass A" is closed, and deleting α from Π_0 yields a prf for $\Gamma^\forall_{\Pi_0} \vdash D$ for any $D \in \Delta_{\Pi_0}$).

Assume Π_2 is the largest such block. Successive lines "close A_1", "ass A_2" can be derived only by $\vee E_1$, from some occurrence $(A_1 \vee A_2) \in \Delta_\Pi$ (or $(A_2 \vee A_1) \in \Delta_\Pi$). We have $\Gamma_{\Pi_0} = (\Gamma_\Pi, A_2)$. By ind.ass. $\Gamma^\forall_{\Pi_0} \vdash C$, and by the case condition $(A_2)^\forall_{\Pi_0} \equiv A_2$; also, for each $F \in \Gamma_{\Pi_0}$, $F^\forall_{\Pi_0} \equiv F^\forall_\Pi$; so

(2) $\quad \Gamma^\forall_\Pi, A_2 \vdash C.$

Subcase 6.1.1. The displayed marker "close A_1" closes a block (with no initial closed sub-block) which, w.l.o.g., is the one corresponding to the inference. Then $\Delta_\Pi = (\Delta_{\Pi_2}, C)$. By the case and subcase conditions, $\Gamma_{\Pi_1} \subset (\Gamma_\Pi, A_1)$, $(A_1)_{\Pi_1} \equiv A_1$, and for each $F \in \Gamma_{\Pi_1}$, $F^\forall_\Pi$ differs from $F^\forall_{\Pi_1}$ only in additional universal closure. By ind.ass.

applied to Π_1 $\Gamma_{\Pi_1} \vdash C$; so

(3) $\Gamma_\Pi , A_1 \vdash C$

Again, by the case condition, $\Gamma_\Pi \supset \Gamma_{\Pi_2}$; by ind.ass. applied to Π_2 $\Gamma^\forall_{\Pi_2} \vdash D$ for each $D \in \Delta_{\Pi_2}$. Also, for each $F \in \Gamma_{\Pi_2}$ $F^\forall_\Pi$ differs from $F^\forall_{\Pi_2}$ by extra universal closure; so

(4) $\Gamma^\forall_\Pi \vdash D$ for each $D \in \Delta_{\Pi_2}$.

To prove (4) also for $D :\equiv C$ we take in (4) $D \equiv A_1 \vee A_2$; with (2) and (3) we get $\Gamma^\forall_\Pi \vdash C$.

Subcase 6.1.2. Otherwise; then $\Delta_\Pi = \Delta_{\Pi_1}$, $\Gamma_\Pi = \Gamma_{\Pi_1}$ and for each $F \in \Gamma_\Pi$ $F^\forall_\Pi$ differs from $F^\forall_{\Pi_1}$ only by universal closure; (1) follows by ind.ass. applied to Π_1.

Case 6.2. Otherwise; then $\Gamma_\Pi = \Gamma_{\Pi_0}$, $\Gamma^\forall_\Pi = \Gamma^\forall_{\Pi_0}$ and $\Delta_\Pi = \Delta_{\Pi_0}$.

(1) follows outright by ind.ass.

7. $\forall I$. $\Pi = \begin{matrix} \Pi_0 \\ \forall x A(x), \text{ for } z \end{matrix}$

We have $\Gamma_\Pi = \Gamma_{\Pi_0}$, $(\Gamma^\forall_\Pi) = (\forall z)\Gamma^\forall_{\Pi_0}$, $\Delta_\Pi = (\Delta_{\Pi_0}, \forall x A(x))$. By ind.ass. $\Gamma^\forall_{\Pi_0} \vdash D$ for each $D \in \Delta_{\Pi_0}$, so $\Gamma_\Pi \vdash D$ for each $D \in \Delta_\Pi$, $D \not\equiv \forall x A(x)$. Also, $A(z) \in \Delta_{\Pi_0}$, so $\Gamma^\forall_\Pi \vdash \forall x A(x)$.

8. $\exists E_1$. $\Pi = \begin{matrix} \Pi_0 \\ \text{ass } A(z), \text{ for } z \end{matrix}$. Trivial.

9. $\exists E_2$. $\Pi = \begin{cases} \begin{matrix} \Pi_1 \\ \Pi_0 \text{ ass } A(z), \text{ for } z \end{matrix} \\ \begin{matrix} \Pi_2 \\ \text{close } A(z), C \end{matrix} \end{cases}$

Case 9.1. The displayed marker "close A(z)" closes a block Π_2 with no initial sub-block; w.l.o.g. this is the largest block associated with the inference (for otherwise the marker closes a lower assumption "ass A(z)", which may be deleted without any effect).

We have $\Gamma_{\Pi_0} = (\Gamma_\Pi, A(z))$, $\Delta_\Pi = (\Delta_{\Pi_1}, C)$, $\exists x A(x) \in \Delta_{\Pi_1}$, and by the case condition $\Gamma_{\Pi_1} \subset \Gamma_\Pi$. By ind.ass. applied to Π_1, $\Gamma^{\forall}_{\Pi_1} \vdash D$ for each $D \in \Delta_{\Pi_1}$; so (as in cases above) $\Gamma^{\forall}_{\Pi} \vdash D$ for $D \in \Delta_\Pi$, $D \not\equiv C$. Taking $D \equiv \exists x A(x)$, we get

(5) $$\Gamma^{\forall}_{\Pi} \vdash \exists x A(x).$$

Also, by ind.ass. applied to Π_0, $\Gamma^{\forall}_{\Pi_0} \vdash C$. But $\Gamma_{\Pi_0} = (\Gamma_\Pi, A(z))$, and by the case condition $A(z)^{\forall}_{\Pi_0} \equiv A(z)$; so

(6) $$\Gamma^{\forall}_{\Pi}, A(z) \vdash C$$

The variable z does not occur free in $\Gamma^{\forall}_{\Pi}$, since it is marked below any assumption in Π; moreover, z is not free in C by assumption. Thus (5) and (6) imply $\Gamma^{\forall}_{\Pi} \vdash C$.

Case 9.2. Otherwise. Then $\Gamma_\Pi = \Gamma_{\Pi_0}$, $\Gamma^{\forall}_{\Pi} = (\forall z)(\Gamma^{\forall}_{\Pi_0})$, $\Delta_\Pi = \Delta_{\Pi_0}$. By ind.ass. $\Gamma^{\forall}_{\Pi_0} \vdash D$ for $D \in \Delta_{\Pi_0}$; so $\Gamma^{\forall}_{\Pi} \vdash D$ for $D \in \Delta_\Pi$. This concludes the proof. ⊠

Remark: An alternative calculus for classical logic. Change the calculus above as follows.

1. Add a second condition to the definition of a prf: no variable is marked more than once.

2. Delete the condition on variable-occurrence in the definition of closed blocks.

3. (optional) Reformulate the rule for existential elimination as $\frac{\exists x A(x)}{A(z),\ \text{for } z}$.

The resulting calculus is complete and sound for classical first-order logic. Using the notations of the Theorem above, each prf Π derives $\Gamma_\Pi^\forall \vdash D_\Pi^\exists$ for each $D \in \Delta_\Pi$, where $D_\Pi^\exists$ is the existential closure of D for all variables marked in Π.

Note that in this calculus there is a proof without the classical rule $\perp_c$, deriving $B \to \exists x Ax . \to \exists x (B \to Ax)$, which is not intuitionistically correct.

4. An example.

To illustrate the difference between the three approaches discussed above, we present a somewhat less trivial tautology, for which we give prfs in each one of the calculi considered. For the block approach we use the variant [C0].

Tautology: $\forall x(A \vee Bx) \wedge \forall x(Bx \to Cx . \wedge . A \to D \wedge Ex) \to . . \forall x(\neg A \to Cx) \wedge . \forall x \neg Bx \to (F \to D) . \wedge \forall x E$

4.1. Sequential approach--core prf

1.	$\forall x(A \vee Bx) \wedge \forall x(Bx\ Cx . \wedge . A \to D \wedge Ex)$	-
2.	$\forall x(A \vee Bx)$	1.
3.	$A \vee Bx$	2.
4.	$\forall x(Bx \to Cx . \wedge . A \to D \wedge Ex)$	1.

5.	$Bx \to Cx . \wedge . A \to D \wedge Ex$	4.
6.	$Bx \to Cx$	5
7.	$A \to D \wedge Ex$	6
8.	A	-
9.	$\neg A$	-
10.	$\perp$	8,9
11.	Cx	10
12.	$\neg A \to Cx$	11
13.	Bx	-
14.	Cx	6,13
15.	$\neg A \to Cx$	14
16.	$\neg A \to Cx$	3,12,15
17.	$\forall x (\neg A \to Cx)$	16
18.	$D \wedge Ex$	7,8
19.	D	18
20.	$F \to D$	19
21.	Ex	18
22.	$\forall x Ex$	21
23.	$F \to D . \wedge . \forall x Ex$	20,22
24.	$\forall x \neg Bx \to . . (F \to D) . \wedge \forall x Ex$	23
25.	$\forall x \neg Bx$	-
26.	$\neg Bx$	25
27.	$\perp$	13,26
28.	$F \to D . \wedge . \forall x Ex$	27
29.	$\forall x \neg Bx \to . F \to D . \wedge . \forall x Ex$	28
30.	$\forall x \neg Bx \to . F \to D . \wedge . \forall x Ex$	3,24,29
31.	$\forall x (\neg A \to Cx) \wedge . \forall x \neg Bx \to (F \to D) \wedge \forall x Ex$	17,30
32.	$\forall x (A \vee Bx) \wedge \forall x (Bx \to Cx . \wedge . A \to D \wedge Ex) \to \forall x (\neg A \to Cx) \wedge . \forall x \neg Bx \to (F \to D) \wedge \forall x Ex$	31

4.2. Edited version of the above.

inactive assumptions	active assumptions		core	premises	remote
	ass.	1.	$\forall x(A\vee Bx)\wedge\forall x(Bx\rightarrow(x.\wedge.A\rightarrow D\wedge Ex)$		
1		2.	$\forall x(A\vee Bx)$	1	
1		3.	$A\vee Bx$	2	
1		4.	$\forall x(Bx\rightarrow(x.\wedge.A\rightarrow D\wedge Ex)$	1	
1		5.	$Bx\rightarrow Cx.\wedge.A\rightarrow D\wedge Ex$	4	
1		6.	$Bx\rightarrow Cx$	5	
1		7.	$A\rightarrow D\wedge Ex$	5	
	ass	8.	A		
	ass	9.	$\neg A$		
8,9		10.	$\perp$	8,9	
8	9 = $\neg A$	11.	Cx	10	
	8 = A	12.	$\neg A\rightarrow Cx$	11	
	ass	13.	Bx		
7,13		14.	Cx	6,13	6 = $Bx\rightarrow Cx$
1	13 = Bx	15.	$\neg A\rightarrow Cx$	14	
1		16.	$\neg A\rightarrow Cx$	3,12,15	3 = $A\vee Bx$
1		17.	$\forall x(\neg A\rightarrow Cx)$	16	
		[8]	A		
1,8		18.	$D\wedge Ex$	7,8	7 = $A\rightarrow D\wedge Ex$
1,8		19.	D	18	
1,8		20.	$F\rightarrow D$	19	
1,8		21.	Ex	18	
1,8		22.	$\forall xEx$	21	
1,8		23.	$(F\rightarrow D)\wedge\forall xEx$	20,22	
1	8 = A	24.	$\forall x\neg Bx\rightarrow(F\rightarrow D)\wedge\forall xEx$	23	
		[13]	Bx		
	ass	25.	$\forall x\neg Bx$		
25		26.	$\neg Bx$	25	
13,25		27.	$\perp$	13,26	
13	25 = $\forall x\neg Bx$	28.	$(F\rightarrow D)\wedge\forall xEx$	27	
	13 = Bx	29.	$\forall x\neg Bx\rightarrow(F\rightarrow D)\wedge\forall xEx$	28	
		30.	$\forall x\neg Bx\rightarrow(F\rightarrow D)\wedge\forall xEx$	3,24,29	3 = $A\vee Bx$
	1 = $\forall x(A\vee Bx)$	31.	$\forall x(\neg A\rightarrow Cx)\wedge.$		
	$\wedge\forall x(Bx\rightarrow Cx.\wedge.A\rightarrow D\wedge Ex)$		$\forall x\neg Bx\rightarrow(F\rightarrow D)\wedge\forall xEx$	17,30	17 = $\forall x(\neg A\rightarrow Cx)$
		32.	$\forall x(A\vee Bx)\wedge\forall x(Bx\rightarrow Cx.\wedge.A\rightarrow D\wedge Ex)$		
			$\rightarrow\forall x(\neg A\rightarrow Cx)\wedge.\ \forall x\neg Bx\rightarrow(F\rightarrow D)\wedge\forall xEx$	31	

4.3. A prf in the calculus of [CO].

```
Proof
        assume ∀x(A∨Bx)∧∀x(Bx→Cx.∧. A→D∧Ex)
        ∀x(A∨Bx)
        ∀x(Bx→Cx.∧.A→D∧Ex)
        Proof
            Arbitrary x
            A∨Bx
            Proof
                Assume A
                Proof
                    Assume ¬A
                    false
                    Cx
                    qed
                ¬A→Cx
            Proof
                Assume Bx
                Proof
                    Assume ¬A
                    Bx→Cx.∧.A→D∧Ex
                    Bx→Cx
                    Cx
                    qed
                ¬A→Cx
                qed
            ¬A→Cx
            qed
        ∀x(¬A→Cx)
        Proof
            Assume ∀x ¬Bx
            A∨Bx
            Proof
                Assume A
                Bx→Cx.∧.A→D∧Ex
                A→D∧Ex
                D∧Ex
                Proof
                    Assume F
                    D
                    qed
                F→D
                qed
            Proof
                Assume Bx
                ¬Bx
                false
                F→D
                qed
```

```
                F→D
                Proof
                    Arbitrary x
                    A∨Bx
                    Proof
                        Assume A
                        Bx→Cx.∧.A→D∧Ex
                        A→D∧Ex
                        D∧Ex
                        Ex
                        qed
                    Proof
                        Assume Bx
                        ¬Bx
                        false
                        Ex
                        qed
                    Ex
                    qed
                ∀xEx
                (F→D)∧∀xEx
                qed
            ∀x¬Bx→(F→D)∧∀xEx
            ∀x(¬A→Dx)∧.∀x¬Bx→(F→D)∧∀xEx
             qed
∀x(A∨Bx)∧∀x(Bx→Cx.∧.A→D∧Ex)→∀x(¬A→Cx)∧.∀x¬Bx→(F→D)∧∀xEx
```

4.4. A prf in the calculus of §3.

```
1.   Ass. ∀x(A∨Bx)∧∀x(Bx→Cx.∧.A→D∧Ex)
2.         ∀x(A∨Bx)
.          A∨Bx
.          ∀x(Bx→Cx.∧.A→D∧Ex)
.          Bx→Cx
           A D Ex
           Ass. A
                Ass ¬A
                     ⊥
                     Cx
                close ¬A
                ¬A→Cx
           close A
           Ass Bx                          [∨E1]
               Cx
               ¬A→Cx
           close Bx
           ¬A→Cx                           [∨E2]
           ∀x(¬A→Cx)
```

```
Ass A
    D∧Ex
    D
    F→D
    Ex
    ∀xEx
    (F→D)∧∀xEx
    ∀x¬Bx→(F→D)∧∀x Ex
close A
Ass Bx                                  [vE1]
    Ass ∀x¬Bx
        ¬Bx

            (F→D)∧∀xEx
    close ∀x¬Bx
    ∀x¬Bx→(F→D)∧∀xEx
Close Bx
∀x¬Bx→(F→D)∧∀xEx                        [vE2]
∀x(¬A→Cx)∧.∀x¬Bx→(F→D)∧∀xEx
Close 1
∀x(A∨Bx)∧∀x(Bx→Cx.∧.A→D∧Ex)→..∀x(¬A→Cx)∧.∀x¬Bx→(F→D)∧∀xEx
```

5. A comparison between the three approachs

We use SND, BND and LND for the sequential, block and local approachs to linear ND, discussed respectively in sections 1,2 and 3 above.

5.1. Conceptual and pedagogical aspects

(1) Simplicity and coherence of the definition.

SND seems to have the simplest definition, using a minimum amount of auxiliary notions. The definition uses sequents, but in a less involved way than an inductive definition for BND (as in 2. above).

As mentioned above, the usual definition for BND by accessibility conditions is not altogether systematic in several respects. The mixed use of fmls and prfs as premises of inferences ensues the mixed use of local and global conditions on local cor-

rectness of prfs. Prfs actually grow both ways (as when $\forall E$ or $\exists E$ is used), in contradiction to the underlying attempt to parallel Hilbert-style calculi. Finally, the application of an inference may consist in inserting delineators, in addition to writing down the conclusion.

The LND approach of § 3 above is systematic in its emphasis on local conditions, on separation between global and local aspects of the proof, and in the growth of prfs exclusively downwards.

(2) Clarity of underlying ideas.

All linear ND calculi have Gentzen's ND as a guiding model, and introduce extra structure to compensate for the effects of linearization. In SND the extra structure--the sequents-- is already implicit in Gentzen's system, and the nature of the original system comes through clearly.

The usual definition of BND is more dissimilar to Gentzen's formulation, and the equivalence of the two variants less perspicious LND is further divorced from Gentzen's system; the need for a proof of soundness of the calculus testifies to the relative obscurity of the underlying ideas in LND.

(3) Structural closure properties.

As for Hilbert-type calculi, in all calculi invented by Gentzen an initial portion of a prf is a prf. This remains true for SND and LND, but not for BND. In the variant of [CO], for instance, delineators such as Proof or Arbitrary x cannot remain in a proof if their peers are deleted.

(4) Conformity with programming style.

One is tempted to regard BND as an analogue to syntactic programming blocks. However, the similarity is only partial. Half a proof-block corresponding, say, to →I, is meaningful, since no termination-step (closing the assumption) is necessary for the assumption statement itself to make sense. By contradistinction, a termination step for a guard-condition to, say, a while-loop, is necessary for the condition to have a meaning, since this meaning is operative, and depends on the global structure of the program (to use a hyperbole, Logic is timeless).

For Programming Logic it is sometimes necessary to put restrictions on the presence or the form of assumptions, although the usual restrictions (on →I, ∨E and ∃E) are indirectly taken care of by the accessibility conditions (cf. [CO]). In BND and--especially--LND such conditions are harmful to the stylistic unity. In contrast, conditions of this type are easily stated and verified in SND.

5.2. Convenience of use

(1) Conciseness and flexibility.

For isomorphic prfs, SND presents a savings over LND, and LND--over BND. With respect to the [CO] variant of BND, LND uses less delineators per (implicit) block, and dispense with delineators for ∀I.

As explained in the introduction, all linear ND calculi often allow one to dispense with repeated display of the same sub-prf (compulsory in the tree variant). The more liberal the structure of the derivations is, the more economy. Thus, in BND the ∀I blocks prohibit a prf like that of § 2, example 1.

(2) Readability.

To verify a correct use of ∨E in BND one has to point one's index at four different prior locations in the prf, arbitrarily apart. To verify just a single inference, one must rely on parsing of a large portion of the prf, which must include all the lines in between the first premise and the conclusion; this becomes a gigantic problem for large prfs.

The situation is slightly better with LND, where conditions are simpler, and various aspects of verification are separated (verification of ∨E necessitates only two prior locations). Also, accessiblity conditions are easy to verify; but then the problem is simply shifted to verifying closure of assumptions. Still, the whole is somewhat simpler than with BND.

SND seems, again, one's best bet. An edited version is easy to read and verify. There is no increase in complexity of local verification as the proof grows longer (the increase resulting from the presence of more inactive assumptions in antecedents is marginal). This contrasts with an exponential increase for BND and, essentially, also for LND.

5.3. Computer implementation.

It is useful to recall that formal proofs have not really taken roots to date in mathematics, although (exactly) one century has passed since the publication of Frege's *Begriffsschrift*. Mathematicians are usually satisfied with the conviction that proofs can be formalized in principle, leaving it to the logicians to refer to formal proofs as objects for study. A calculus sufficiently concise and flexible to be widely used at least for the hard core of Mathematics has yet to be invented.

Programmers and computer scientists should be expected to be at least as pragmatic and skeptical as the mathematicians, and to accept a rigid logical formalism only if it offers definite advantages over semi-formal comments inserted in the programs. The gain may be either in increased clarity, precision and conciseness, or in the possibility of obtaining substantial machine assistance.

The core version of SND seems to offer the most from the first, subjective, point of view. As for machine assistance, all formal calculi permit automatic proof checking; however, SND allows, in addition, the use of the machine to complete a short sketch into a fully edited proof.

Moreover, SND is the simplest to use and implement in a computer environment. The verification algorithms are simple, and debugging of proofs is easier than in other alternatives. This makes it more attractive to add an SND verifier and editor to existing compilers. SND is also attractive for top-down presentation, and for automated theorem proving; it is thus a style suitable in environments requiring both automated proving and automated verification.

There remains the pedagogical question of actually instructing the calculus to programmers. We believe that such an instruction should begin, in any event, by an exposition of Gentzen's original tree-format, at least for the positive fragment of the language (i.e., without disjunction and existential quantification). Once this is done, passing to SND seems the most natural step, and the dichotomy of fmls vs. sequents would be fully understood, and fully reflected in the pragmatic dichotomy of core version vs. expanded edited version. The opportunity for the student to use the machine to complete his sketchy proof might be an additional incentive.

6. Built-in derived rules

It often seems a picky nuisance to have to formally derive obvious and frequently-occurring inferences, just as it would be to multiply by hand 43 by 100,000 using the standard multiplication algorithm. In [CO] it is suggested to allow certain fmls as "immediate consequences" of others. The problem is to draw the line: what is immediate to one may be obscure to another. A moment of thought is necessary to verify for the first time even a trivial rule like $\frac{A\wedge B\rightarrow C}{A\rightarrow(B\rightarrow C)}$; on the other hand, any serious text in Intuitionistic Mathematics would take for granted $\frac{\neg\neg\forall xA}{\forall x\neg\neg A}$, which for the non-initiated looks just as good as the intuitionistically unsound rule $\frac{\forall x\neg\neg A}{\neg\neg\forall xA}$.

We propose a syntactic criterion for built-in derived rules. Let us say that a binary connective binds twice, and a unary op-

erator--once, where the unary operators are $\neg, \forall, \exists$ and $\neg\neg$.
The logical depth of a fml is defined as usual: $d(A)=0$ for A atomic; $d(\forall xA)=d(\exists xA)=d(\neg A)=d(\neg\neg A)=d(A)+1$; $d(A\wedge B)=d(A\vee B)=d(A\to B)=\max(d(A),d(B))+1$. For a pair of fmls we define the depth to be that of their conjunction.

Convention: A derived inference rule is immediate if the premise(s) have up to 4 bindings, and depth ≤ 2; and the same holds for the conclusion.

In the following list of immediate inference-rules we indicate the presence of trivial variants by "etc.".

1. $\frac{A\wedge B}{B\wedge A}$; $\frac{A\wedge.B\wedge C}{A\wedge B.\wedge C}$; $\frac{A\vee B}{B\vee A}$; $\frac{A\vee.B\vee C}{A\vee B.\vee C}$; $\frac{A\vee.B\wedge C}{A\vee B}$ etc.

2. $A\to(B\to A)$; $A\ A$;
$\frac{A\to B\ \ B\to C}{A\to C}$; $\frac{A\ \ B\to C}{(A\to B)\to C}$ etc.; $\frac{A\to B}{\neg B\to\neg A}$, $\frac{A\to\neg B}{B\to\neg A}$; $\frac{A\to B\ \neg B}{\neg A}$ etc.

$\frac{\neg(A\to B)}{\neg B}$; $\frac{A}{\neg\neg A}$

classical system only: $\frac{A\to B}{\neg B\to\neg A}$; $\frac{\neg\neg A}{A}$; $\neg\neg A\to A$; $\frac{\neg(A\to B)}{A}$ etc.

intuitionistic system only: $\frac{\neg\neg\neg A}{\neg A}$; $\frac{A\to B}{\neg\neg A\to\neg\neg B}$ etc.; $\frac{\neg\neg(A\to B)}{\neg\neg A\to\neg\neg B}$ etc.;

$\frac{\neg(A\to B)}{\neg\neg A}$ etc; $\frac{\neg\neg A\ \ A\to B}{\neg\neg B}$ etc.

3. $A\wedge B\to A$ etc.; $A\to A\vee B$ etc.

$\frac{\neg(A\vee B)}{\neg A}$ etc.; $\frac{\neg A\ \ \neg B}{\neg(A\vee B)}$ etc.; $\frac{\neg A}{\neg(A\wedge B)}$ etc. $\frac{A\vee B\ \ \neg B}{A}$ etc.; $\frac{\neg(A\wedge B)\ \ A}{\neg B}$ etc

classical system only: $A\vee\neg A$, $\frac{\neg(A\wedge B)}{\neg A\vee\neg B}$

intuitionistic system only: $\frac{\neg\neg(A\wedge B)}{\neg\neg A}$ etc.; $\frac{\neg\neg A\ \ \neg\neg B}{\neg\neg(A\wedge B)}$ etc. $\frac{\neg\neg A\vee\neg\neg B}{\neg\neg(A\vee B)}$ etc.

4. $\frac{A\wedge B\to C}{A\to(B\to C)}$; $\frac{A\to(B\to C)}{A\wedge B\to C}$; $\frac{A\to B \quad A\to C}{A\to B\wedge C}$; $\frac{A\vee B\to C}{A\to C}$ etc.; $\frac{A\to C \quad B\to C}{A\vee B\to C}$; $\frac{\neg A\vee B}{A\to B}$

For the classical system: $\frac{A\to B}{\neg A\vee B}$

5. $\frac{\forall x\neg Ax}{\neg\exists xAx}$; $\frac{\neg\exists xAx}{\forall x\neg Ax}$; $\frac{\exists x\neg Ax}{\neg\forall xAx}$

For the classical system: $\frac{\neg\forall xAx}{\exists x\neg Ax}$

For the intuitionistic system: $\frac{\neg\neg\forall xAx}{\forall x\neg\neg Ax}$; $\frac{\exists x\neg\neg Ax}{\neg\neg\exists xAx}$

6. $\frac{\forall x(Ax\to Bx)\ \forall xAx}{\forall xBx}$ etc.; $\frac{\forall x(Ax\to Bx)\ \exists xAx}{\exists xBx}$ etc.(in particular: $\frac{\forall x(Ax\to}{\exists xAx\to\exists}$

$\frac{\exists xAx\to B}{\forall x(Ax\to B)}$; $\frac{\exists x(Ax\to Bx)\,\forall xAx}{\exists xBx}$ etc.; $\frac{A\to\forall xBx}{\forall x(A\to Bx)}$; $\frac{A\to B(t)}{A\to\exists xB(x)}$ etc.; $\frac{A(t)\to B}{\forall xA(x)\to B}$

For the classical system: $\frac{\forall xAx\to B}{\exists x(Ax\to B)}$; $\frac{A\to\exists xBx.}{\exists x(A\to Bx)}$

7. $\frac{\forall x(Ax\wedge Bx)}{\forall xAx}$ etc.; $\frac{\forall xAx\ \forall xBx}{\forall x(Ax\wedge Bx)}$ etc. $\frac{A\wedge\forall xBx}{\forall x(A\vee Bx)}$ etc. (x not free in A

$\frac{\exists x(Ax\wedge Bx)}{\exists xAx}$ etc; $\frac{\exists xAx\ \forall xBx}{\exists x(Ax\wedge Bx)}$ etc; $\frac{A\ \exists xBx}{\exists x(A\wedge Bx)}$; $\frac{A\vee\exists xBx}{\exists x(A\vee Bx)}$; $\frac{\exists x(Ax\vee Bx)}{\exists xAx\vee\exists xBx}$;

For the classical system: $\frac{\forall x(A\vee Bx)}{A\vee\forall xBx}$ (x not free in A).

Examples. (1) The following uses derived rules from the list above to infer concisely a portion of the example of § 4.

1.	$(A\vee B)\wedge(B\to C)$	ass
2.	$A\vee B$	1
3.	$\neg A\to B$	2
4.	$B\to C$	1
5.	$\neg A\to C$	3,4
6.	$(A\vee B)\wedge(B\to C)\to(\neg A\to C)$	5(1)

(2) We derive intuitionistically $\neg\neg\forall xA\rightarrow\neg\exists x\neg A$.

1.	$\exists x\neg A$	ass
2.	$\neg\forall xA$	1
3.	$\exists x\neg A\rightarrow\neg\forall xA$	2(1)
4.	$\neg\neg\forall xA\rightarrow\neg\exists x\neg A$	3

7. Alternatives to Natural Deduction

After inventing ND, Gentzen discovered (to his own surprise) his Sequent Calculus (SC). What distinguishes SC from ND is not the explicit use of sequents, but a symmetry between antececents and succedents of sequents. In ND one has introduction and elimination rules for each logical constant, which, in the sequential representation, operate on succedents only. By contrast, SC contains only introduction rules, but of two kinds for each constant: on the antecedent and on the succedent (plus certain structural rules that do not involve the logical constants).

SC may be thought of as a meta-calculus for ND, in the sense that inference rules act as instructions for constructing ND prfs (cf. [Pr 65] pp. 90-91, [Zu]). In Proof Theory SC has become the dominant style, since it presents a number of advantages. E.g., the characterization of classical logic is more convenient than in ND; the delineation of the intuitionistic sub-system is purely structural; the (intuitionistic) rules for $\vee$ and $\exists$ are in line with the other inference rules; complexity of fmls in prfs grows only downwords; "normal" (= "cut-free") prfs are characterized by a local condition, making easier their use in applications.

However, it seems that ND is more appropriate than SC for practical implementation. The merit of SC as a meta-calculus for ND is of little value, as one is usually giving a prf, not instructions for constructing one. As a logical calculus, SC does not reflect informal and intuitive forms of argumentation as well as does ND. This is especially true of programming logic for any programming language of Von Neumann style (not so much for LISP or Functional Programming). For such languages the flow of execution is explicit in the syntax of the program; hence, reasoning within the program is more suitable in a "time-oriented" formalism, like ND, where one proceeds from assumptions to conclusions, rather than in SC, where assumptions and conclusions play a symmetric role w.r.t. the flow of argument.

An argument can be made that the left-introduction rules (i.e, rules for introducing logical constants in the antecedents) are convenient and natural. E.g., $\frac{\Gamma, A \Rightarrow C \quad \Gamma, B \Rightarrow C}{\Gamma, A \vee B \Rightarrow C}$. Let us recall, however, that all such rules appear in the list of derived rules given in §6, where the rule above for $\rightarrow L$ is represented by $\frac{A \rightarrow C \quad B \rightarrow C}{A \vee B \rightarrow C}$ (in #4). The formula-format of these rules is at least as appealing for actual argumentation as their SC counterparts.

K. Schütte has invented yet other types of logical calculi. One such calculus focuses on positive and negative parts of fmls, is restricted to classical logic, and is a world apart from informal reasoning in general, and from constructive reasoning in particular, (cf. [Sch 77]). Another calculus of Schütte's ([Sch 6
uses a representation of sequents by fmls, and necessitates in practice cumbersome notational conventions which make the system impractical for implementation.

Rerefences

[CO] R. Constable and M. O'Donnell: A Programming Logic; Winthrop, Cambridge, Mass., 1978.

[Pr 65] D. Prawitz: Natural Deduction; Almqvist & Wiksell, Stockholm, 1965.

[Pr 71] D. Prawitz: Ideas and results in proof theory; in Fenstad (ed.): Proc. of the Second Scandinavian Logic Symp., North Holland, Amsterdam, 1971.

[Sch 62] K. Schütte: Der Interpolationssatz der intuitionistischen Prädikatenlogik; Math. Annalen 148 (1962) 192-200.

[Sch 77] K. Schütte: Proof Theory (2nd ed.), Springer-Verlag, Berlin, 1977.

[Zu] J. Zucker: Cut elimination and normalization; Annals Math. Logic 7 (1974) 1-112.