CHAPTER 1

Logic

The main subject of Mathematical Logic is mathematical proof. In this introductory chapter we deal with the basics of formalizing such proofs. The system we pick for the representation of proofs is Gentzen’s natural deduction, from [5]. Our reasons for this choice are twofold. First, as the name says this is a *natural* notion of formal proof, which means that the way proofs are represented corresponds very much to the way a careful mathematician writing out all details of an argument would go anyway. Second, formal proofs in natural deduction are closely related (via the so-called Curry-Howard correspondence) to terms in typed lambda calculus. This provides us not only with a compact notation for logical derivations (which otherwise tend to become somewhat unmanagable tree-like structures), but also opens up a route to applying the computational techniques which underpin lambda calculus.

Apart from classical logic we will also deal with more constructive logics: minimal and intuitionistic logic. This will reveal some interesting aspects of proofs, e.g. that it is possible und useful to distinguish between existential proofs that actually construct witnessing objects, and others that don’t. As an example, consider the following proposition.

There are irrational numbers $a, b$ such that $a^b$ is rational.

This can be proved as follows, by cases.

Case $\sqrt{2}^{\sqrt{2}}$ is rational. Choose $a = \sqrt{2}$ and $b = \sqrt{2}$. Then $a, b$ are irrational and by assumption $a^b$ is rational.

Case $\sqrt{2}^{\sqrt{2}}$ is irrational. Choose $a = \sqrt{2}^{\sqrt{2}}$ and $b = \sqrt{2}$. Then by assumption $a, b$ are irrational and

$$a^b = \left(\sqrt{2}^{\sqrt{2}}\right)^{\sqrt{2}} = \left(\sqrt{2}\right)^2 = 2$$

is rational.

As long as we have not decided whether $\sqrt{2}^{\sqrt{2}}$ is rational, we do not know which numbers $a, b$ we must take. Hence we have an example of an existence proof which does not provide an instance.

An essential point for Mathematical Logic is to fix a formal language to be used. We take implication $\rightarrow$ and the universal quantifier $\forall$ as basic. Then the logic rules correspond to lambda calculus. The additional connectives $\bot, \exists, \lor$ and $\land$ are defined via axiom schemes. These axiom schemes will later be seen as special cases of introduction and elimination rules for inductive definitions.
1. Formal Languages

1.1. Terms and Formulas. Let a countable infinite set \( \{ v_i \mid i \in \mathbb{N} \} \) of variables be given; they will be denoted by \( x, y, z \). A first order language \( \mathcal{L} \) then is determined by its signature, which is to mean the following.

- For every natural number \( n \geq 0 \) a (possible empty) set of \( n \)-ary relation symbols (also called predicate symbols). 0-ary relation symbols are called propositional symbols. \( \bot \) (read “false”) is required as a fixed propositional symbol. The language will not, unless stated otherwise, contain \( = \) as a primitive.
- For every natural number \( n \geq 0 \) a (possible empty) set of \( n \)-ary function symbols. 0-ary function symbols are called constants.

We assume that all these sets of variables, relation and function symbols are disjoint.

For instance the language \( \mathcal{L}_G \) of group theory is determined by the signature consisting of the following relation and function symbols: the group operation \( \circ \) (a binary function symbol), the unit \( e \) (a constant), the inverse operation \( ^{-1} \) (a unary function symbol) and finally equality \( = \) (a binary relation symbol).

\( \mathcal{L} \)-terms are inductively defined as follows.

- Every variable is an \( \mathcal{L} \)-term.
- Every constant of \( \mathcal{L} \) is an \( \mathcal{L} \)-term.
- If \( t_1, \ldots, t_n \) are \( \mathcal{L} \)-terms and \( f \) is an \( n \)-ary function symbol of \( \mathcal{L} \) with \( n \geq 1 \), then \( f(t_1, \ldots, t_n) \) is an \( \mathcal{L} \)-term.

From \( \mathcal{L} \)-terms one constructs \( \mathcal{L} \)-prime formulas, also called atomic formulas of \( \mathcal{L} \). If \( t_1, \ldots, t_n \) are terms and \( R \) is an \( n \)-ary relation symbol of \( \mathcal{L} \), then \( R(t_1, \ldots, t_n) \) is an \( \mathcal{L} \)-prime formula.

\( \mathcal{L} \)-formulas are inductively defined from \( \mathcal{L} \)-prime formulas by

- Every \( \mathcal{L} \)-prime formula is an \( \mathcal{L} \)-formula.
- If \( A \) and \( B \) are \( \mathcal{L} \)-formulas, then so are \( (A \rightarrow B) \) (“if \( A \), then \( B \)”), \( (A \land B) \) (“\( A \) and \( B \)”) and \( (A \lor B) \) (“\( A \) or \( B \)”).
- If \( A \) is an \( \mathcal{L} \)-formula and \( x \) is a variable, then \( \forall x A \) (“for all \( x \), \( A \) holds”) and \( \exists x A \) (“there is an \( x \) such that \( A \)”) are \( \mathcal{L} \)-formulas.

Negation, classical disjunction, and the classical existential quantifier are defined by

\[
\neg A \ := \ A \rightarrow \bot,
\]
\[
A \land B \ := \ \neg A \rightarrow \neg B \rightarrow \bot,
\]
\[
\exists x A \ := \ \neg \forall x \neg A.
\]

Usually we fix a language \( \mathcal{L} \), and speak of terms and formulas instead of \( \mathcal{L} \)-terms and \( \mathcal{L} \)-formulas. We use

- \( \tau, \sigma, \tau \) for terms,
- \( x, y, z \) for variables,
- \( c \) for constants,
- \( P, Q, R \) for relation symbols,
- \( f, g, h \) for function symbols,
- \( A, B, C, D \) for formulas.
DEFINITION. The depth \( dp(A) \) of a formula \( A \) is the maximum length of a branch in its construction tree. In other words, we define recursively \( dp(P) = 0 \) for atomic \( P \), \( dp(A \circ B) = \max(dp(A), dp(B)) + 1 \) for binary operators \( \circ \), \( dp(\circ A) = dp(A) + 1 \) for unary operators \( \circ \).

The size or length \( |A| \) of a formula \( A \) is the number of occurrences of logical symbols and atomic formulas (parentheses not counted) in \( A \): \( |P| = 1 \) for \( P \) atomic, \( |A \circ B| = |A| + |B| + 1 \) for binary operators \( \circ \), \( |\circ A| = |A| + 1 \) for unary operators \( \circ \).

One can show easily that \( |A| + 1 \leq 2^{dp(A)+1} \).

NOTATION (Saving on parentheses). In writing formulas we save on parentheses by assuming that \( \forall, \exists, \neg \) bind more strongly than \( \land, \lor \), and that in turn \( \land, \lor \) bind more strongly than \( \rightarrow, \leftrightarrow \) (where \( A \leftrightarrow B \) abbreviates \( (A \rightarrow B) \land (B \rightarrow A) \)). Outermost parentheses are also usually dropped. Thus \( A \land \neg B \rightarrow C \) is read as \( ((A \land (\neg B)) \rightarrow C) \). In the case of iterated implications we sometimes use the short notation

\[
A_1 \rightarrow A_2 \rightarrow \ldots A_{n-1} \rightarrow A_n \quad \text{for} \quad A_1 \rightarrow (A_2 \rightarrow \ldots (A_{n-1} \rightarrow A_n) \ldots).
\]

To save parentheses in quantified formulas, we use a mild form of the dot notation: a dot immediately after \( \forall x \) or \( \exists x \) makes the scope of that quantifier as large as possible, given the parentheses around. So \( \forall x.A \rightarrow B \) means \( \forall x.(A \rightarrow B) \), not \( (\forall x.A) \rightarrow B \).

We also save on parentheses by writing e.g. \( Rxyz \), \( Rtz_1t_2 \) instead of \( R(x, y, z) \), \( R(t_0, t_1, t_2) \), where \( R \) is some predicate symbol. Similarly for a unary function symbol with a (typographically) simple argument, so \( f x \) for \( f(x) \), etc. In this case no confusion will arise. But readability requires that we write in full \( R(fx, gy, hz) \), instead of \( Rfxgyhz \).

Binary function and relation symbols are usually written in infix notation, e.g. \( x + y \) instead of \( +(x, y) \), and \( x < y \) instead of \( <(x, y) \). We write \( t \neq s \) for \( \neg(t = s) \) and \( t < s \) for \( \neg(t < s) \).

1.2. Substitution, Free and Bound Variables. Expressions \( \mathcal{E}, \mathcal{E}' \) which differ only in the names of bound variables will be regarded as identical. This is sometimes expressed by saying that \( \mathcal{E} \) and \( \mathcal{E}' \) are \( \alpha \)-equivalent. In other words, we are only interested in expressions “modulo renaming of bound variables”. There are methods of finding unique representatives for such expressions, for example the namefree terms of de Bruijn [4]. For the human reader such representations are less convenient, so we shall stick to the use of bound variables.

In the definition of ‘substitution of expression \( \mathcal{E}' \) for variable \( x \) in expression \( \mathcal{E}'' \)’, either one requires that no variable free in \( \mathcal{E}' \) becomes bound by a variable-binding operator in \( \mathcal{E} \), when the free occurrences of \( x \) are replaced by \( \mathcal{E}' \) (also expressed by saying that there must be no “clashes of variables”), “\( \mathcal{E}' \) is free for \( x \) in \( \mathcal{E}'' \)”, or the substitution operation is taken to involve a systematic renaming operation for the bound variables, avoiding clashes. Having stated that we are only interested in expressions modulo renaming bound variables, we can without loss of generality assume that substitution is always possible.
Also, it is never a real restriction to assume that distinct quantifier occurrences are followed by distinct variables, and that the sets of bound and free variables of a formula are disjoint.

**NOTATION.** “FV” is used for the (set of) free variables of an expression; so FV(t) is the set of variables free in the term t, FV(A) the set of variables free in formula A etc.

\[ \mathcal{E}[x := t] \] denotes the result of substituting the term t for the variable x in the expression \( \mathcal{E} \). Similarly, \( \mathcal{E}[\vec{x} := \vec{t}] \) is the result of simultaneously substituting the terms \( \vec{t} = t_1, \ldots, t_n \) for the variables \( \vec{x} = x_1, \ldots, x_n \), respectively.

Locally we shall adopt the following convention. In an argument, once a formula has been introduced as \( A(x) \), i.e., \( A \) with a designated variable \( x \), we write \( A(t) \) for \( A[x := t] \), and similarly with more variables. □

### 1.3. Subformulas

Unless stated otherwise, the notion of subformula we use will be that of a subformula in the sense of Gentzen.

**DEFINITION.** (Gentzen) subformulas of \( A \) are defined by

(a) \( A \) is a subformula of \( A \);
(b) if \( B \circ C \) is a subformula of \( A \) then so are \( B, C \), for \( \circ = \rightarrow, \land, \lor \);
(c) if \( \forall x B \) or \( \exists x B \) is a subformula of \( A \), then so is \( B[x := t] \), for all \( t \) free for \( x \) in \( B \).

If we replace the third clause by:

(c') if \( \forall x B \) or \( \exists x B \) is a subformula of \( A \) then so is \( B \),
we obtain the notion of literal subformula.

**DEFINITION.** The notions of positive, negative, strictly positive subformula are defined in a similar style:

(a) \( A \) is a positive and a strictly positive subformula of itself;
(b) if \( B \land C \) or \( B \lor C \) is a positive [negative, strictly positive] subformula of \( A \), then so are \( B, C \);
(c) if \( \forall x B \) or \( \exists x B \) is a positive [negative, strictly positive] subformula of \( A \), then so is \( B[x := t] \);
(d) if \( B \rightarrow C \) is a positive [negative] subformula of \( A \), then \( B \) is a negative [positive] subformula of \( A \), and \( C \) is a positive [negative] subformula of \( A \);
(e) if \( B \rightarrow C \) is a strictly positive subformula of \( A \), then so is \( C \).

A strictly positive subformula of \( A \) is also called a strictly positive part \((s.p.p.)\) of \( A \). Note that the set of subformulas of \( A \) is the union of the positive and negative subformulas of \( A \). Literal positive, negative, strictly positive subformulas may be defined in the obvious way by restricting the clause for quantifiers.

**EXAMPLE.** \( (P \rightarrow Q) \rightarrow R \land \forall x R'(x) \) has as \( s.p.p.'s \) the whole formula, \( R \land \forall x R'(x) \), \( R \land \forall x R'(x) \), \( R(t) \). The positive subformulas are the \( s.p.p.'s \) and in addition \( P \); the negative subformulas are \( P \rightarrow Q, Q \).

### 2. Natural Deduction

We introduce Gentzen’s system of natural deduction. To allow a direct correspondence with the lambda calculus, we restrict the rules used to those
for the logical connective $\rightarrow$ and the universal quantifier $\forall$. The rules come in pairs: we have an introduction and an elimination rule for each of these.

The other logical connectives are introduced by means of axiom schemes: this is done for conjunction $\land$, disjunction $\lor$ and the existential quantifier $\exists$. The resulting system is called minimal logic; it has been introduced by Johansson in 1937 [9]. Notice that no negation is present.

If we then go on and require the ex-falso-quodlibet scheme for the nullary propositional symbol $\bot$ ("falsum"), we can embed intuitionistic logic. To obtain classical logic, we add as an axiom scheme the principle of indirect proof, also called stability. However, to obtain classical logic it suffices to restrict to the language based on $\rightarrow$, $\lor$, $\land$ and $\forall$; we can introduce classical disjunction $\lor^c$ and the classical existential quantifier $\exists^c$ via their (classical) definitions above. For these the usual introduction and elimination properties can then be derived.

### 2.1. Examples of Derivations

Let us start with some examples for natural proofs. Assume that a first order language $\mathcal{L}$ is given. For simplicity we only consider here proofs in pure logic, i.e. without assumptions (axioms) on the functions and relations used.

1. $(A \rightarrow B \rightarrow C) \rightarrow (A \rightarrow B) \rightarrow A \rightarrow C$

Assume $A \rightarrow B \rightarrow C$. To show: $(A \rightarrow B) \rightarrow A \rightarrow C$. So assume $A \rightarrow B$. To show: $A \rightarrow C$. So finally assume $A$. To show: $C$. We have $A$, by the last assumption. Hence also $B \rightarrow C$, by the first assumption, and $B$, using the next to last assumption. From $B \rightarrow C$ and $B$ we obtain $C$, as required.

2. $(\forall x . A \rightarrow B) \rightarrow A \rightarrow \forall x B$, if $x \notin \text{FV}(A)$.

Assume $\forall x . A \rightarrow B$. To show: $A \rightarrow \forall x B$. So assume $A$. To show: $\forall x B$. Let $x$ be arbitrary; note that we have not made any assumptions on $x$. To show: $B$. We have $A \rightarrow B$, by the first assumption. Hence also $B$, by the second assumption.

3. $(A \rightarrow \forall x B) \rightarrow \forall x . A \rightarrow B$, if $x \notin \text{FV}(A)$.

Assume $A \rightarrow \forall x B$. To show: $\forall x . A \rightarrow B$. Let $x$ be arbitrary; note that we have not made any assumptions on $x$. To show: $A \rightarrow B$. So assume $A$. To show: $B$. We have $\forall x B$, by the first and second assumption. Hence also $B$.

A characteristic feature of these proofs is that assumptions are introduced and eliminated again. At any point in time during the proof the free or "open" assumptions are known, but as the proof progresses, free assumptions may become cancelled or "closed" because of the implies-introduction rule.

We now reserve the word proof for the informal level; a formal representation of a proof will be called a derivation.

An intuitive way to communicate derivations is to view them as labelled trees. The labels of the inner nodes are the formulas derived at those points, and the labels of the leaves are formulas or terms. The labels of the nodes immediately above a node $\nu$ are the premises of the rule application, the formula at node $\nu$ is its conclusion. At the root of the tree we have the conclusion of the whole derivation. In natural deduction systems one works
with assumptions affixed to some leaves of the tree; they can be open or else closed.

Any of these assumptions carries a marker. As markers we use assumption variables \( \Box_0, \Box_1, \ldots \), denoted by \( u, v, w, u_0, u_1, \ldots \). The (previous) variables will now often be called object variables, to distinguish them from assumption variables. If at a later stage (i.e. at a node below an assumption) the dependency on this assumption is removed, we record this by writing down the assumption variable. Since the same assumption can be used many times (this was the case in example (1)), the assumption marked with \( u \) (and communicated by \( u: A \)) may appear many times. However, we insist that distinct assumption formulas must have distinct markers.

An inner node of the tree is understood as the result of passing form premises to a conclusion, as described by a given rule. The label of the node then contains in addition to the conclusion also the name of the rule. In some cases the rule binds or closes an assumption variable \( u \) (and hence removes the dependency of all assumptions \( u: A \) marked with that \( u \)). An application of the \( \forall \)-introduction rule similarly binds an object variable \( x \) (and hence removes the dependency on \( x \)). In both cases the bound assumption or object variable is added to the label of the node.

2.2. Introduction and Elimination Rules for \( \rightarrow \) and \( \forall \). We now formulate the rules of natural deduction. First we have an assumption rule, that allows an arbitrary formula \( A \) to be put down, together with a marker \( u \):

\[
\begin{array}{c}
u: A \text{ Assumption} \\
\end{array}
\]

The other rules of natural deduction split into introduction rules (I-rules for short) and elimination rules (E-rules) for the logical connectives \( \rightarrow \) and \( \forall \). For implication \( \rightarrow \) there is an introduction rule \( \rightarrow^+ u \) and an elimination rule \( \rightarrow^- \), also called modus ponens. The left premise \( A \rightarrow B \) in \( \rightarrow^- \) is called major premise (or main premise), and the right premise \( A \) minor premise (or side premise). Note that with an application of the \( \rightarrow^+ u \)-rule all assumptions above it marked with \( u: A \) are cancelled.

\[
\frac{[u: A]}{\begin{array}{c}
\begin{array}{c}
B \\
\end{array}
\end{array}}
\begin{array}{c}
A \rightarrow B \\
\end{array}
\rightarrow^+ u
\]

\[
\frac{\begin{array}{c}
M \\
\end{array}}{\begin{array}{c}
\begin{array}{c}
A \rightarrow B \\
\end{array}
\end{array} \rightarrow^-}
\]

For the universal quantifier \( \forall \) there is an introduction rule \( \forall^+ x \) and an elimination rule \( \forall^- \), whose right premise is the term \( r \) to be substituted.

The rule \( \forall^+ x \) is subject to the following (Eigen-) variable condition: The derivation \( M \) of the premise \( A \) should not contain any open assumption with \( x \) as a free variable.

\[
\begin{array}{c}
\begin{array}{c}
M \\
\end{array}
\end{array}
\frac{A}{\forall x A} \forall^+ x
\]

\[
\begin{array}{c}
\begin{array}{c}
M \\
\end{array}
\end{array}
\frac{\forall x A}{A[x := r]} \forall^-
\]

We now give derivations for the example formulas (1) – (3). Since in many cases the rule used is determined by the formula on the node, we
suppress in such cases the name of the rule,

\[
\begin{array}{c}
u: A \to B \to C \quad w: A \quad v: A \to B \quad w: A \\
\hline
B \to C \\
\hline
A \to C \to^+ w \\
(A \to B) \to A \to C \to^+ v \\
(A \to B \to C) \to (A \to B) \to A \to C \to^+ u
\end{array}
\]  

Note here that the variable condition is satisfied: \(x\) is not free in \(A\) (and also not free in \(\forall x. A \to B\)).

\[
\begin{array}{c}
u: A \to \forall x B \quad x \\
\hline
\forall x B \\
\hline
A \to B \to^+ v \\
\forall x . A \to B \to^+ x \\
(A \to \forall x B) \to \forall x . A \to B \to^+ u
\end{array}
\]  

Here too the variable condition is satisfied: \(x\) is not free in \(A\).

### 2.3. Axiom Schemes for Disjunction, Conjunction, Existence and Falsity

We follow the usual practice of considering all free variables in an axiom as universally quantified outside.

**Disjunction.** The introduction axioms are

\[
\forall^+_0: A \rightarrow A \lor B \\
\forall^+_1: B \rightarrow A \lor B
\]

and the elimination axiom is

\[
\lor^-: (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow A \lor B \rightarrow C.
\]

**Conjunction.** The introduction axiom is

\[
\land^+: A \rightarrow B \rightarrow A \land B
\]

and the elimination axiom is

\[
\land^-: (A \rightarrow B \rightarrow C) \rightarrow A \land B \rightarrow C.
\]

**Existential Quantifier.** The introduction axiom is

\[
\exists^+: A \rightarrow \exists x A
\]

and the elimination axiom is

\[
\exists^-: (\forall x . A \rightarrow B) \rightarrow \exists x A \rightarrow B \quad (x \text{ not free in } B).
\]
\textit{Falsity.} This example is somewhat extreme, since there is no introduction axiom; the elimination axiom is
\[ \bot^-: \bot \rightarrow A. \]
In the literature this axiom is frequently called “ex-falso-quoilibet”, written \( \text{Eq} \). It clearly is derivable from its instances \( \bot \rightarrow R \xi \), for every relation symbol \( R \).

\textit{Equality.} The introduction axiom is
\[ \text{Eq}^+: \text{Eq}(x, x) \]
and the elimination axiom is
\[ \text{Eq}^-: \forall x R(x, x) \rightarrow \text{Eq}(x, y) \rightarrow R(x, y). \]
It is an easy exercise to show that the usual equality axioms can be derived.

All these axioms can be seen as special cases of a general scheme, that of an \textit{inductively defined predicate}, which is defined by some introduction rules and one elimination rule. We will study this kind of definition in full generality in Chapter 6. \( \text{Eq}(x, y) \) is a binary such predicate, \( \bot \) is a nullary one, and \( A \lor B \) another nullary one which however depends on the two parameter predicates \( A \) and \( B \).

The desire to follow this general pattern is also the reason that we have chosen our rather strange \( \land^- \)-axiom, instead of the more obvious \( A \land B \rightarrow A \) and \( A \land B \rightarrow B \) (which clearly are equivalent).

\section*{2.4. Minimal, Intuitionistic and Classical Logic.}
We write \( \vdash A \) and call \( A \) \textit{derivable} (in \textit{minimal logic}), if there is a derivation of \( A \) without free assumptions, from the axioms of 2.3 using the rules from 2.2, but \textit{without using the ex-falso-quoilibet axiom, i.e., the elimination axiom} \( \bot^- \) for \( \bot \). A formula \( B \) is called derivable from assumptions \( A_1, \ldots, A_n \), if there is a derivation (without \( \bot^- \)) of \( B \) with free assumptions among \( A_1, \ldots, A_n \). Let \( \Gamma \) be a (finite or infinite) set of formulas. We write \( \Gamma \vdash B \) if the formula \( B \) is derivable from finitely many assumptions \( A_1, \ldots, A_n \in \Gamma \).

Similarly we write \( \vdash_i A \) and \( \Gamma \vdash_i B \) if use of the ex-falso-quoilibet axiom is allowed; we then speak of derivability in \textit{intuitionistic logic}.

For classical logic there is no need to use the full set of logical connectives: classical disjunction as well as the classical existential quantifier are defined, by \( A \lor B := \neg A \rightarrow B \rightarrow \bot \) and \( \exists^c x A := \forall x \neg \neg A \). Moreover, when dealing with derivability we can even get rid of conjunction; this can be seen from the following lemma:

\textbf{Lemma (Elimination of} \( \land \)). \textit{For each formula} \( A \) \textit{built with the connectives} \( \rightarrow, \land, \lor, \forall \) \textit{there are formulae} \( A_1, \ldots, A_n \) \textit{without} \( \land \) \textit{such that} \( \vdash A \leftrightarrow \bigwedge_{i=1}^n A_i \).

\textbf{Proof.} Induction on \( A \). \textbf{Case} \( R \xi \). Take \( n = 1 \) and \( A_1 := R \xi \). \textbf{Case} \( A \land B \). By induction hypothesis, we have \( A_1, \ldots, A_n \) and \( B_1, \ldots, B_m \). Take \( A_1, \ldots, A_n, B_1, \ldots, B_m \). \textbf{Case} \( A \rightarrow B \). By induction hypothesis, we have \( A_1, \ldots, A_n \) and \( B_1, \ldots, B_m \). For the sake of notational simplicity assume \( n = 2 \) and \( m = 3 \). Then
\[ \vdash (A_1 \land A_2 \rightarrow B_1 \land B_2 \land B_3) \]
\[ \leftrightarrow (A_1 \to A_2 \to B_1) \land (A_1 \to A_2 \to B_2) \land (A_1 \to A_2 \to B_3). \]

**Case \( \forall x A \).** By induction hypothesis for \( A \), we have \( A_1, \ldots, A_n \). Take \( \forall x A_1, \ldots, \forall x A_n \), for
\[ \vdash \forall x \bigwedge_{i=1}^{n} A_i \leftrightarrow \bigwedge_{i=1}^{n} \forall x A_i. \]
This concludes the proof.

For the rest of this section, let us restrict to the language based on \( \to \), \( \forall \), \( \bot \) and \( \land \). We obtain classical logic by adding, for every relation symbol \( R \) distinct from \( \bot \), the principle of indirect proof expressed as the so-called "stability axiom" (\( \text{Stab}_R \)):
\[ \neg \neg R \overline{x} \to R \overline{x}. \]

Let
\[ \text{Stab} := \{ \forall \overline{x}. \neg \neg R \overline{x} \to R \overline{x} \mid R \text{ relation symbol distinct from } \bot \}. \]

We call the formula \( A \) classically derivable and write \( \vdash_c A \) if there is a derivation of \( A \) from stability assumptions \( \text{Stab}_R \). Similarly we define classical derivability from \( \Gamma \) and write \( \Gamma \vdash_c A \), i.e.
\[ \Gamma \vdash_c A \iff \Gamma \cup \text{Stab} \vdash A. \]

**Theorem (Stability, or Principle of Indirect Proof).** For every formula \( A \) (of our language based on \( \to \), \( \forall \), \( \bot \) and \( \land \)),
\[ \vdash_c \neg \neg A \to A. \]

**Proof.** Induction on \( A \). For simplicity, in the derivation to be constructed we leave out applications of \( \to^+ \) at the end. **Case \( R \overline{\Gamma} \) with \( R \) distinct from \( \bot \).** Use \( \text{Stab}_R \). **Case \( \bot \).** Observe that \( \neg \bot \to \bot = ((\bot \to \bot) \to \bot) \to \bot \). The desired derivation is
\[
\begin{array}{c}
v : (\bot \to \bot) \to \bot \\
u : \bot \to \bot \\
\bot \\
u \vdash u
\end{array}
\]

**Case \( A \to B \).** Use \( \vdash (\neg B \to B) \to \neg \neg (A \to B) \to A \to B \); a derivation is
\[
\begin{array}{c}
u_1 : \neg B \\
u_2 : A \to B \\
w : A
\end{array}
\]
\[
\begin{array}{c}
u_1 \vdash B \\
u_2 \vdash (A \to B) \\
(\bot \vdash \neg (A \to B)) \vdash^+ w_2 \\
u_1 \vdash \bot \\
u_2 \vdash \neg B \to B \\
u_2 \vdash B \\
u_1 \vdash \neg B \\
u_2 \vdash \neg B \to + u_1
\end{array}
\]

**Case \( \forall x A \).** Clearly it suffices to show \( \vdash (\neg \neg A \to A) \to \neg \neg \forall x A \to A \); a derivation is
\[
\begin{array}{c}
u_1 : \neg A \\
u_2 : \forall x A \\
x
\end{array}
\]
\[
\begin{array}{c}
u_1 \vdash A \\
u_2 \vdash \forall x A \\
x \vdash A \\
u_1 \vdash \bot \\
u_2 \vdash \neg \forall x A \\
(\bot \vdash \neg \forall x A) \vdash^+ w_2 \\
u_1 \vdash \bot \\
u_2 \vdash \neg \forall x A \\
u_1 \vdash \bot \\
u_2 \vdash \neg \forall x A \to + u_1 \\
u_1 \vdash \bot \\
u_2 \vdash \neg \forall x A \to + u_1 \\
u_1 \vdash \bot \\
u_2 \vdash \neg \forall x A \to + u_1 \\
u_1 \vdash \bot \\
u_2 \vdash \neg \forall x A \to + u_1 \\
u_1 \vdash \bot \\
u_2 \vdash \neg \forall x A \to + u_1 \\
u_1 \vdash \bot \\
u_2 \vdash \neg \forall x A \to + u_1 \\
u_1 \vdash \bot \\
u_2 \vdash \neg \forall x A \to + u_1 \\
u_1 \vdash \bot \\
u_2 \vdash \neg \forall x A \to + u_1 \\
u_1 \vdash \bot \\
u_2 \vdash \neg \forall x A \to + u_1 \\
u_1 \vdash \bot \\
u_2 \vdash \neg \forall x A \to + u_1 \\
u_1 \vdash \bot \\
u_2 \vdash \neg \forall x A \to + u_1 \\
u_1 \vdash \bot \\
u_2 \vdash \neg \forall x A \to + u_1 \\
u_1 \vdash \bot \\
u_2 \vdash \neg \forall x A \to + u_1 \\
u_1 \vdash \bot \\
u_2 \vdash \neg \forall x A \to + u_1 \\
u_1 \vdash \bot \\
u_2 \vdash \neg \forall x A \to + u_1 \\
u_1 \vdash \bot \\
u_2 \vdash \neg \forall x A \to + u_1 \\
u_1 \vdash \bot \\
u_2 \vdash \neg \forall x A \to + u_1 \\
u_1 \vdash \bot \\
u_2 \vdash \neg \forall x A \to + u_1 \\
u_1 \vdash \bot \\
u_2 \vdash \neg \forall x A \to + u_1 \\
u_1 \vdash \bot \\
u_2 \vdash \neg \forall x A \to + u_1 \\
u_1 \vdash \bot \\
u_2 \vdash \neg \forall x A \to + u_1 \\
u_1 \vdash \bot \\
u_2 \vdash \neg \forall x A \to + u_1 \\
u_1 \vdash \bot \\
u_2 \vdash \neg \forall x A \to + u_1 \\
u_1 \vdash \bot \\
u_2 \vdash \neg \forall x A \to + u_1 \\
u_1 \vdash \bot \\
The case $A \land B$ is left to the reader. 

Notice that clearly $\vdash_c \bot \to A$, for stability is stronger:

$$\dfrac{M_{\text{Stab}} \quad u : \bot \to \neg \neg A \quad \neg \neg A \to A \quad \neg \neg A}{\bot \to A \to \neg \neg A \to \neg \neg A}$$

where $M_{\text{Stab}}$ is the (classical) derivation of stability.

Notice also that even for the $\to$, $\bot$-fragment the inclusion of minimal logic in intuitionistic logic, and of the latter in classical logic are proper. Examples are

$$\vdash \bot \to P, \quad \text{but} \quad \vdash_\iota \bot \to P, \quad \vdash \neg \neg ((P \to Q) \to P) \to P, \quad \text{but} \quad \vdash_c ((P \to Q) \to P) \to P.$$ 

Non-derivability can be proved by means of countermodels, using a semantic characterization of derivability; this will be done in Chapter 2. $\vdash \iota \bot \to P$ is obvious, and the Peirce formula $((P \to Q) \to P) \to P$ can be derived in minimal logic from $\bot \to Q$ and $\neg \neg P \to P$, hence is derivable in classical logic.

2.5. Negative Translation. We embed classical logic into minimal logic, via the so-called negative (or Gödel-Gentzen) translation.

A formula $A$ is called negative, if every atomic formula of $A$ distinct from $\bot$ occurs negated, and $A$ does not contain $\forall$, $\exists$.

**Lemma.** For negative $A$, $\vdash \neg \neg A \to A$.

**Proof.** This follows from the proof of the stability theorem, using $\vdash \neg \neg R^{\iota} \to \neg \neg R^{\iota}$. 

Since $\forall$, $\exists$ do not occur in formulas of classical logic, in the rest of this section we consider the language based on $\to$, $\forall$, $\bot$ and $\land$ only.

**Definition** (Negative translation $^g$ of Gödel-Gentzen).

$$(R^{\iota})^g : = \neg \neg R^{\iota} \quad \text{(R distinct from $\bot$)}$$

$$(\bot)^g : = \bot, \quad (A \land B)^g : = A^g \land B^g, \quad (A \to B)^g : = A^g \to B^g, \quad (\forall x A)^g : = \forall x A^g.$$ 

**Theorem.** For all formulas $A$,

(a) $\vdash_c A \iff A^g$,

(b) $\Gamma \vdash_c A \iff \Gamma^g \vdash A^g$, where $\Gamma^g := \{ B^g \mid B \in \Gamma \}$.

**Proof.** (a) The claim follows from the fact that $\vdash_c$ is compatible with equivalence. 2. $\iff$. Obvious $\Rightarrow$. By induction on the classical derivation. For a stability assumption $\neg \neg R^{\iota} \to R^{\iota}$ we have $\neg \neg R^{\iota} \to R^{\iota}$. 


\[ \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \nega
3.1. Conversion. A conversion eliminates a detour in a derivation, i.e., an elimination immediately following an introduction. We consider the following conversions:

$$
\begin{align*}
\text{→-conversion.} \\
[u : A] & | N \\
M & | N \\
\frac{B}{A \to B} & \to^+ u | A \\
\frac{A \to B}{B} & A \to^- \\
\end{align*}
$$

$$
\forall\text{-conversion.} \\
M & | M' \\
\frac{\forall x A}{\forall^+ x} & \forall^- \\
\frac{A[x := r]}{A[x := r]} & A \to^- \\
\end{align*}
$$

3.2. Derivations as Terms. It will be convenient to represent derivations as terms, where the derived formula is viewed as the type of the term. This representation is known under the name Curry-Howard correspondence.

We give an inductive definition of derivation terms in the table below, where for clarity we have written the corresponding derivations to the left. For the universal quantifier $\forall$ there is an introduction rule $\forall^+ x$ and an elimination rule $\forall^-$, whose right premise is the term $r$ to be substituted. The rule $\forall^+ x$ is subject to the following (Eigen-) variable condition: The derivation term $M$ of the premise $A$ should not contain any open assumption with $x$ as a free variable.

3.3. Reduction, Normal Form. Although every derivation term carries a formula as its type, we shall usually leave these formulas implicit and write derivation terms without them.

Notice that every derivation term can be written uniquely in one of the forms

$$ uM | \lambda v M | (\lambda v M) N \bar{L}, $$

where $u$ is an assumption variable or assumption constant, $v$ is an assumption variable or object variable, and $M$, $N$, $\bar{L}$ are derivation terms or object terms.

Here the final form is not normal: $(\lambda v M) N \bar{L}$ is called $\beta$-redex (for “reducible expression”). The conversion rule is

$$ (\lambda v M) N \to_\beta M[v := N]. $$

Notice that in a substitution $M[v := N]$ with $M$ a derivation term and $v$ an object variable, one also needs to substitute in the formulas of $M$.

The closure of the conversion relation $\to_\beta$ is defined by

- If $M \to_\beta M'$, then $M \to M'$.
- If $M \to M'$, then also $MN \to M'N$, $NM \to NM'$, $\lambda v M \to \lambda v M'$ (inner reductions).

So $M \to N$ means that $M$ reduces in one step to $N$, i.e., $N$ is obtained from $M$ by replacement of (an occurrence of) a redex $M'$ of $M$ by a conversion $M''$ of $M'$, i.e. by a single conversion. The relation $\to^+$ ("properly
### Table 1. Derivation terms for \( \rightarrow \) and \( \forall \)

<table>
<thead>
<tr>
<th>derivation</th>
<th>term</th>
</tr>
</thead>
<tbody>
<tr>
<td>( u : A )</td>
<td>( u^A )</td>
</tr>
<tr>
<td>[ u : A ]</td>
<td>( (\lambda u^A M^B)^A \rightarrow B )</td>
</tr>
<tr>
<td>[ M ]</td>
<td>( B )</td>
</tr>
<tr>
<td>( A \rightarrow B \rightarrow^+ u )</td>
<td></td>
</tr>
<tr>
<td>(</td>
<td>M ) ( \frac{N}{A} \rightarrow B \rightarrow A \rightarrow )</td>
</tr>
<tr>
<td>(</td>
<td>M ) ( \frac{\forall x.A}{A} \rightarrow x ) (with var.cond.)</td>
</tr>
<tr>
<td>(</td>
<td>M ) ( \frac{\forall x.A}{A[x := r]} \rightarrow ) ( \forall^− )</td>
</tr>
</tbody>
</table>

A term \( M \) is in normal form, or \( M \) is normal, if \( M \) does not contain a redex. \( M \) has a normal form if there is a normal \( N \) such that \( M \rightarrow^* N \).

A reduction sequence is a (finite or infinite) sequence \( M_0 \rightarrow M_1 \rightarrow M_2 \ldots \) such that \( M_i \rightarrow M_{i+1} \), for all \( i \).

Finite reduction sequences are partially ordered under the initial part relation; the collection of finite reduction sequences starting from a term \( M \) forms a tree, the reduction tree of \( M \). The branches of this tree may be identified with the collection of all infinite and all terminating finite reduction sequences.

A term is strongly normalizing if its reduction tree is finite.

**Example.**

\[
(\lambda x\lambda y\lambda z.x(z(yz)))(\lambda u\lambda v u)(\lambda u'\lambda v' u') \rightarrow \\
(\lambda y\lambda z.(\lambda u\lambda v u)z(yz))(\lambda u'\lambda v' u') \rightarrow
\]
\[(\lambda y \lambda z.(\lambda w z)(yz))(\lambda u^t \lambda v^t u^t) \rightarrow \\]
\[(\lambda y \lambda z z)(\lambda u^t \lambda v^t u^t) \rightarrow \lambda z z.\]

**Lemma (Substitutivity of \(\rightarrow\)).** (a) If \(M \rightarrow M'\), then \(MN \rightarrow M'N\).
(b) If \(N \rightarrow N'\), then \(MN \rightarrow MN'\).
(c) If \(M \rightarrow M'\), then \(M[v := N] \rightarrow M'[v := N]\).
(d) If \(N \rightarrow N'\), then \(M[v := N] \rightarrow^* M[v := N']\).

**Proof.** (a) and (c) are proved by induction on \(M \rightarrow M'\); (b) and (d) by induction on \(M\). Notice that the reason for \(\rightarrow^*\) in (d) is the fact that \(v\) may have many occurrences in \(M\). \(\square\)

**3.4. Strong Normalization.** We show that every term is strongly normalizing.

To this end, define by recursion on \(k\) a relation \(sn(M,k)\) between terms \(M\) and natural numbers \(k\) with the intention that \(k\) is an upper bound on the number of reduction steps up to normal form.

\[sn(M,0) \iff M \text{ is in normal form},\]
\[sn(M, k + 1) \iff sn(M',k) \text{ for all } M' \text{ such that } M \rightarrow M'.\]

Clearly a term is strongly normalizable if there is a \(k\) such that \(sn(M,k)\). We first prove some closure properties of the relation \(sn\).

**Lemma (Properties of \(sn\)).** (a) If \(sn(M,k)\), then \(sn(M,k+1)\).
(b) If \(sn(MN,k)\), then \(sn(M,k)\).
(c) If \(sn(M_i, k_i)\) for \(i = 1 \ldots n\), then \(sn(uM_1 \ldots M_n, k_1 + \ldots + k_n)\).
(d) If \(sn(M,k)\), then \(sn(\lambda v M,k)\).
(e) If \(sn(M[v := N]^L,k)\) and \(sn(N,l)\), then \(sn((\lambda v M)^N^L,k + l + 1)\).

**Proof.** (a). Induction on \(k\). Assume \(sn(M,k)\). We show \(sn(M,k+1)\). So let \(M'\) with \(M \rightarrow M'\) be given; because of \(sn(M,k)\) we must have \(k > 0\).

We have to show \(sn(M',k)\). Because of \(sn(M,k)\) we have \(sn(M',k-1)\), hence by induction hypothesis \(sn(M',k)\).

(b). Induction on \(k\). Assume \(sn(MN,k)\). We show \(sn(M,k)\). In case \(k = 0\) the term \(MN\) is normal, hence also \(M\) is normal and therefore \(sn(M,0)\).

So let \(k > 0\) and \(M \rightarrow M'\); we have to show \(sn(M',k-1)\). From \(M \rightarrow M'\) we have \(MN \rightarrow M'N\). Because of \(sn(MN,k)\) we have by definition \(sn(M'N,k-1)\), hence \(sn(M',k-1)\) by induction hypothesis.

(c). Assume \(sn(M_i,k_i)\) for \(i = 1 \ldots n\). We show \(sn(uM_1 \ldots M_n,k)\) with \(k := k_1 + \ldots + k_n\). Again we employ induction on \(k\). In case \(k = 0\) all \(M_i\) are normal, hence also \(uM_1 \ldots M_n\). So let \(k > 0\) and \(uM_1 \ldots M_n \rightarrow M'\). Then \(M' = uM_1 \ldots M_n' \ldots M_n\) with \(M_i \rightarrow M_i'\); we have to show \(sn(uM_1 \ldots M_n',k-1)\). Because of \(M_i \rightarrow M_i'\) and \(sn(M_i,k_i)\) we have \(k_i > 0\) and \(sn(M_i',k_i-1)\), hence \(sn(uM_1 \ldots M_n',k-1)\) by induction hypothesis.

(d). Assume \(sn(M,k)\). We have to show \(sn(\lambda v M,k)\). Use induction on \(k\). In case \(k = 0\) \(M\) is normal, hence \(\lambda v M\) is normal, hence \(sn(\lambda v M,0)\). So let \(k > 0\) and \(\lambda v M \rightarrow L\). Then \(L\) has the form \(\lambda v M'\) with \(M \rightarrow M'\). So \(sn(M',k-1)\) by definition, hence \(sn(\lambda v M',k)\) by induction hypothesis.

(e). Assume \(sn(M[v := N]^L,k)\) and \(sn(N,l)\). We need to show that \(sn((\lambda v M)^N^L,k + l + 1)\). We use induction on \(k + l\). In case \(k + l = 0\) the
3. Normalization

Theorem (Properties of SN). For all formulas $A$, derivation terms $M \in SN$ and $N^A \in SN$ the following holds.

(a) $M[v := N] \in SN$.

(a') $M[x := r] \in SN$.

(b) Suppose $M$ derives $A \rightarrow B$. Then $MN \in SN$.

(b') Suppose $M$ derives $\forall x A$. Then $M\tau \in SN$.

Proof. By course-of-values induction on $dp(A)$, with a side induction on $M \in SN$. Let $N^A \in SN$. We distinguish cases on the form of $M$.

Case $u\tilde{M}$ by (Var) from $\tilde{M} \in SN$. (a). The SIH(a) (SIH means side induction hypothesis) yields $M_i[v := N] \in SN$ for all $M_i$ from $\tilde{M}$. In case $u \neq v$ we immediately have $(u\tilde{M})[v := N] \in SN$. Otherwise we need $N\tilde{M}[v := N] \in SN$. But this follows by multiple applications of IH(b), since every $M_i[v := N]$ derives a subformula of $A$ with smaller depth. (a'). Similar, and simpler. (b), (b'). Use (Var) again.
Case $\lambda v M$ by ($\lambda$) from $M \in SN$. (a), (a'). Use ($\lambda$) again. (b). Our goal is $(\lambda v M)N \in SN$. By ($\beta$) it suffices to show $M[v := N] \in SN$ and $N \in SN$. The latter holds by assumption, and the former by SIH(a); (b'). Similar, and simpler.

Case $(\lambda v M)K\tilde{L}$ by ($\beta$) from $M[w := K]\tilde{L} \in SN$ and $K \in SN$. (a). The SIH(a) yields $M[v := N][w := K[v := N]]\tilde{L}[v := N] \in SN$ and $K[v := N] \in SN$, hence $(\lambda v M[v := N])K[v := N]\tilde{L}[v := N] \in SN$ by ($\beta$). (a'). Similar, and simpler. (b), (b'). Use ($\beta$) again.

COROLLARY. For every term we have $M \in SN$; in particular every term $M$ is strongly normalizable.

PROOF. Induction on the (first) inductive definition of derivation terms $M$. In cases $u$ and $\lambda v M$ the claim follows from the definition of SN, and in case $MN$ it follows from the preceding theorem.

3.5. Confluence. A relation $R$ is said to be confluent, or to have the Church-Rosser property (CR), if, whenever $M_0 R M_1$ and $M_0 R M_2$, then there is an $M_3$ such that $M_1 R M_3$ and $M_2 R M_3$. A relation $R$ is said to be weakly confluent, or to have the weak Church-Rosser property (WCR), if, whenever $M_0 R M_1, M_0 R M_2$ then there is an $M_3$ such that $M_1 R^* M_3$ and $M_2 R^* M_3$, where $R^*$ is the reflexive and transitive closure of $R$.

Clearly for a confluent reduction relation $\rightarrow^*$ the normal forms of terms are unique.

LEMMA (Newman 1942). Let $\rightarrow^*$ be the transitive and reflexive closure of $\rightarrow$, and let $\rightarrow$ be weakly confluent. Then the normal form w.r.t. $\rightarrow$ of a strongly normalizing $M$ is unique. Moreover, if all terms are strongly normalizing w.r.t. $\rightarrow$, then the relation $\rightarrow^*$ is confluent.

PROOF. Call $M$ good if it satisfies the confluence property w.r.t. $\rightarrow^*$, i.e. if whenever $K \leftarrow^* M \rightarrow^* L$, then $K \rightarrow^* N \leftarrow^* L$ for some $N$. We show that every strongly normalizing $M$ is good, by transfinite induction on the well-founded partial order $\rightarrow^+$, restricted to all terms occurring in the reduction tree of $M$. So let $M$ be given and assume

$$\forall M'. M \rightarrow^+ M' \implies M' \text{ is good.}$$

We must show that $M$ is good, so assume $K \leftarrow^* M \rightarrow^* L$. We may further assume that there are $M', M''$ such that $K \leftarrow^* M' \leftarrow M \rightarrow M'' \rightarrow^* L$, for otherwise the claim is trivial. But then the claim follows from the assumed weak confluence and the induction hypothesis for $M'$ and $M''$, as shown in the picture below.

3.6. Uniqueness of Normal Forms. We first show that $\rightarrow$ is weakly confluent. From this and the fact that it is strongly normalizing we can easily infer (using Newman's Lemma) that the normal forms are unique.

PROPOSITION. $\rightarrow$ is weakly confluent.

PROOF. Assume $N_0 \leftarrow M \rightarrow N_1$. We show that $N_0 \rightarrow^* N \leftarrow^* N_1$ for some $N$, by induction on $M$. If there are two inner reductions both on the same subterm, then the claim follows from the induction hypothesis using substitutivity. If they are on distinct subterms, then the subterms do not
overlap and the claim is obvious. It remains to deal with the case of a head reduction together with an inner conversion.

\[ M \vdash (\lambda x. M) \bar{N} \bar{L} \]
\[ M[\bar{v}] := N \bar{L} \]
\[ M'[\bar{v}] := N \bar{L} \]
\[ M'[\bar{v}] := N \bar{L} \]
\[ M[\bar{v}] := N' \bar{L} \]
\[ M[\bar{v}] := N' \bar{L} \]
\[ M[\bar{v}] := N' \bar{L} \]

where for the lower left arrows we have used substitutivity again.

\[ \square \]

**COROLLARY.** Every term is strongly normalizing, hence normal forms are unique.

\[ \square \]

**3.7. The Structure of Normal Derivations.** Let \( M \) be a normal derivation, viewed as a proof tree. A sequence of f.o.'s (formula occurrences) \( A_0, \ldots, A_n \) such that (1) \( A_0 \) is a top formula (leaf) of the proof tree, and (2) for \( 0 \leq i < n \), \( A_{i+1} \) is immediately below \( A_i \), and (3) \( A_i \) is not the minor premise of an \( \to^- \)-application, is called a track of the deduction tree \( M \). A track of order 0 ends in the conclusion of \( M \); a track of order \( n + 1 \) ends in the minor premise of an \( \to^- \)-application with major premise belonging to a track of order \( n \).

Since by normality an E-rule cannot have the conclusion of an I-rule as its major premise, the E-rules have to precede the I-rules in a track, so the following is obvious: a track may be divided into an E-part, say \( A_0, \ldots, A_{i-1} \), a minimal formula \( A_i \), and an I-part \( A_{i+1}, \ldots, A_n \). In the E-part all rules
are E-rules; in the I-part all rules are I-rules; $A_i$ is the conclusion of an E-rule and, if $i < n$, a premise of an I-rule. It is also easy to see that each f.o. of $M$ belongs to some track. Tracks are pieces of branches of the tree with successive f.o.’s in the subformula relationship: either $A_{i+1}$ is a subformula of $A_i$ or vice versa. As a result, all formulas in a track $A_0, \ldots, A_n$ are subformulas of $A_0$ or of $A_n$; and from this, by induction on the order of tracks, we see that every formula in $M$ is a subformula either of an open assumption or of the conclusion. To summarize, we have seen:

**Lemma.** In a normal derivation each formula occurrence belongs to some track.

**Proof.** By induction on the height of normal derivations. 

**Theorem.** In a normal derivation each formula is a subformula of either the end formula or else an assumption formula.

**Proof.** We prove this for tracks of order $n$, by induction on $n$. 

3.8. Normal Versus Non-Normal Derivations. We now show that the requirement to give a normal derivation of a derivable formula can sometimes be unrealistic. Following Statman [16] and Orevkov [13] we give examples of formulas $C_k$ which are easily derivable with non-normal derivations (whose number of nodes is linear in $k$), but which require a non-elementary (in $k$) number of nodes in any normal derivation.

The example is related to Gentzen’s proof in [6] of transfinite induction up to $\omega_k$ in arithmetic. There the function $y \oplus \omega^x$ plays a crucial role, and also the assignment of a “lifting”-formula $A^+(x)$ to any formula $A(x)$, by

$$A^+(x) := \forall y, (\forall z < y) A(z) \rightarrow (\forall z < y \oplus \omega^x) A(z).$$

Here we consider the numerical function $y + 2^x$ instead, and axiomatizes its graph by means of Horn clauses. The formula $C_k$ expresses that from these axioms the existence of $2_k$ follows. A short, non-normal proof of this fact can then be given by a modification of Gentzen’s idea, and it is easily seen that any normal proof of $C_k$ must contain at least $2_k$ nodes.

The derivations to be given make heavy use of the existential quantifier $\exists^d$ defined by $\neg \forall$. In particular we need:

**Lemma (Existence Introduction).** $\vdash A \rightarrow \exists^d x A$.

**Proof.** $\lambda u^A \lambda v^{\forall x \neg A} v . v u$.

**Lemma (Existence Elimination).** $\vdash (\neg \neg B \rightarrow B) \rightarrow \exists^d x A \rightarrow (\forall x . A \rightarrow B) \rightarrow B$ if $x \notin FV(B)$.

**Proof.** $\lambda u^{\neg \neg B \rightarrow B} \lambda v^{\forall x \neg A} \lambda w^{\forall x} A \rightarrow B . u \lambda u_2^B . w \lambda x . \lambda u_1^A . u_2 (w . u x_1)$.

Note that the stability assumption $\neg \neg B \rightarrow B$ is not needed if $B$ does not contain an atom $\neq \bot$ as a strictly positive subformula. This will be the case for the derivations below, where $B$ will always be a classical existential formula.

Let us now fix our language. We use a ternary relation symbol $R$ to represent the graph of the function $y + 2^x$; so $R(y, x, z)$ is intended to mean $y + 2^x = z$. We now axiomatize $R$ by means of Horn clauses. For simplicity we use a unary function symbol $s$ (to be viewed as the successor function)
and a constant 0; one could use logic without function symbols instead – as Orevkov does –, but this makes the formulas somewhat less readable and the proofs less perspicuous. Note that Orevkov’s result is an adaption of a result of Statman [16] for languages containing function symbols.

\[ \text{Hyp}_1: \forall y R(y, 0, s(y)) \]
\[ \text{Hyp}_2: \forall y, x, z, z_1. R(y, x, z) \rightarrow R(z, x, z_1) \rightarrow R(y, s(x), z_1) \]

The goal formula then is

\[ C_k := \exists z_{k}, \ldots, z_0. R(0, 0, z_k) \land R(0, z_k, z_{k-1}) \land \ldots \land R(0, z_1, z_0). \]

To obtain the short proof of the goal formula \( C_k \) we use formulas \( A_i(x) \) with a free parameter \( x \).

\[ A_0(x) := \forall y \exists z R(y, x, z), \]
\[ A_{i+1}(x) := \forall y. A_i(y) \rightarrow \exists z. A_i(z) \land R(y, x, z). \]

For the two lemmata to follow we give an informal argument, which can easily be converted into a formal proof. Note that the existence elimination lemma is used only with existential formulas as conclusions. Hence it is not necessary to use stability axioms and we have a derivation in minimal logic.

\[ \text{LEMMA.} \vdash \text{Hyp}_1 \rightarrow \text{Hyp}_2 \rightarrow A_i(0). \]

\[ \text{PROOF.} \] \textbf{Case} \( i = 0 \). Obvious by \text{Hyp}_1.

\textbf{Case} \( i = 1 \). Let \( x \) with \( A_0(x) \) be given. It is sufficient to show \( A_0(s(x)) \), that is \( \forall y \exists z_1 R(y, s(x), z_1) \). So let \( y \) be given. We know

\[ A_0(x) = \forall y \exists z R(y, x, z). \]

Applying (4) to our \( y \) gives \( z \) such that \( R(y, x, z) \). Applying (4) again to this \( z \) gives \( z_1 \) such that \( R(z, x, z_1) \). By \text{Hyp}_2 we obtain \( R(y, s(x), z_1) \).

\textbf{Case} \( i + 2 \). Let \( x \) with \( A_{i+1}(x) \) be given. It suffices to show \( A_{i+1}(s(x)) \), that is \( \forall y. A_i(y) \rightarrow \exists z. A_i(z) \land R(y, s(x), z) \). So let \( y \) with \( A_i(y) \) be given. We know

\[ A_{i+1}(x) = \forall y. A_i(y) \rightarrow \exists z_1. A_i(z_1) \land R(y, x, z_1). \]

Applying (5) to our \( y \) gives \( z \) such that \( A_i(z) \) and \( R(y, x, z) \). Applying (5) again to this \( z \) gives \( z_1 \) such that \( A_i(z_1) \) and \( R(z, x, z_1) \). By \text{Hyp}_2 we obtain \( R(y, s(x), z_1) \).

\[ \quad \square \]

Note that the derivations given have a fixed length, independent of \( i \).

\[ \text{LEMMA.} \vdash \text{Hyp}_1 \rightarrow \text{Hyp}_2 \rightarrow C_k. \]

\[ \text{PROOF.} \] \( A_k(0) \) applied to 0 and \( A_{k-1}(0) \) yields \( z_k \) with \( A_{k-1}(z_k) \) such that \( R(0, 0, z_k) \).

\( A_{k-1}(z_k) \) applied to 0 and \( A_{k-2}(0) \) yields \( z_{k-1} \) with \( A_{k-2}(z_{k-1}) \) such that \( R(0, z_k, z_{k-1}) \).

\( A_1(z_2) \) applied to 0 and \( A_0(0) \) yields \( z_1 \) with \( A_0(z_1) \) such that \( R(0, z_2, z_1) \).

\( A_0(z_1) \) applied to 0 yields \( z_0 \) with \( R(0, z_1, z_0) \).

\[ \quad \square \]

Note that the derivations given have length linear in \( k \).

We want to compare the length of this derivation of \( C_k \) with the length of an arbitrary normal derivation.
PROPOSITION. Any normal derivation of $C_k$ from Hyp$_1$ and Hyp$_2$ has at least $2^k$ nodes.

PROOF. Let a normal derivation $M$ of falsity $\bot$ from Hyp$_1$, Hyp$_2$ and the additional hypothesis

$$u : \forall z_k, \ldots, z_0. R(0, 0, z_k) \rightarrow R(0, z_k, z_{k-1}) \rightarrow \cdots \rightarrow R(0, z_1, z_0) \rightarrow \bot$$

be given. We may assume that $M$ does not contain free object variables (otherwise substitute them by 0). The main branch of $M$ must begin with $u$, and its side premises are all of the form $R(0, s^n(0), s^k(0))$. Observe that any normal derivation of $R(s^m(0), s^n(0), s^k(0))$ from Hyp$_1$, Hyp$_2$ and $u$ has at least $2^n$ occurrences of Hyp$_1$ and is such that $k = m + 2^n$. This can be seen easily by induction on $n$. Note also that such a derivation cannot involve $u$.

If we apply this observation to the above derivations of the side premises we see that they derive

$$R(0, 0, s^{2^0}(0)), \quad R(0, s^{2^0}(0), s^{2^0}(0)), \quad \ldots \quad R(0, s^{2^{k-1}}(0), s^{2^k}(0)).$$

The last of these derivations uses at least $2^{2k-1} = 2_k$-times Hyp$_1$. \hfill \Box

4. Normalization including Permutative Conversions

The elimination of "detours" done in Section 3 will now be extended to the full language. However, incorporation of $\lor$, $\wedge$ and $\exists$ leads to difficulties. If we do this by means of axioms (or constant derivation terms, as in 2.3), we cannot read off as much as we want from a normal derivation. If we do it in the form of rules, we must also allow permutative conversion. The reason for the difficulty is that in the elimination rules for $\lor$, $\wedge$, $\exists$ the minor premise reappears in the conclusion. This gives rise to a situation where we first introduce a logical connective, then do not touch it (by carrying it along in minor premises of $\lor^-, \wedge^-, \exists^-$), and finally eliminate the connective. This is not a detour as we have treated them in Section 3, and the conversion introduced there cannot deal with this situation. What has to be done is a permutative conversion: permute an elimination immediately following an $\lor^-, \wedge^-, \exists^-$-rule over this rule to the minor premise.

We will show that any sequence of such conversion steps terminates in a normal form, which in fact is uniquely determined (again by Newman’s lemma).

Derivations in normal form have many pleasant properties, for instance:

**Subformula property:** every formula occurring in a normal derivation is a subformula of either the conclusion or else an assumption;

**Explicit definability:** a normal derivation of a formula $\exists x A$ from assumptions not involving disjunctive of existential strictly positive parts ends with an existence introduction, hence also provides a term $r$ and a derivation of $A[x := r]$;

**Disjunction property:** a normal derivation of a disjunction $A \lor B$ from assumptions not involving disjunctions as strictly positive parts ends with a disjunction introduction, hence also provides either a derivation of $A$ or else one of $B$;
4.1. Rules for \( \lor, \land \) and \( \exists \). Notice that we have not given rules for the connectives \( \lor, \land \) and \( \exists \). There are two reasons for this omission:

- They can be covered by means of appropriate axioms as constant derivation terms, as given in 2.3;
- For simplicity we want our derivation terms to be pure lambda terms formed just by lambda abstraction and application. This would be violated by the rules for \( \lor, \land \) and \( \exists \), which require additional constructs.

However – as just noted – in order to have a normalization theorem with a useful subformula property as a consequence we do need to consider rules for these connectives. So here they are:

**Disjunction.** The introduction rules are

\[
\begin{array}{c|c|c}
| & M & M \\
\hline
A & A \lor B & B \\
\hline
\end{array}
\]

and the elimination rule is

\[
[u : A] \quad [v : B] \\
| M & N & K \\
\hline
A \lor B & C & C \\
\hline
C & \lor\lnot u, v
\]

**Conjunction.** The introduction rule is

\[
\begin{array}{c|c|c}
| & M & N \\
\hline
A & A \land B & B \\
\hline
\end{array}
\]

and the elimination rule is

\[
[u : A] \quad [v : B] \\
| M & N \\
\hline
A \land B & C \\
\hline
C & \land\lnot u, v
\]

**Existential Quantifier.** The introduction rule is

\[
\begin{array}{c|c}
| M \\
\hline
r & A[x := r] \\
\hline
\end{array}
\]

and the elimination rule is

\[
[u : A] \\
| M & N \\
\hline
\exists x A & B \\
\hline
B & \exists\lnot x, u \text{ (var.cond.)}
\]

The rule \( \exists\lnot x, u \) is subject to the following *(Eigen-)* variable condition: The derivation \( N \) should not contain any open assumptions apart from \( u : A \) whose assumption formula contains \( x \) free, and moreover \( B \) should not contain the variable \( x \) free.

It is easy to see that for each of the connectives \( \lor, \land, \exists \) the rules and the axioms are equivalent, in the sense that from the axioms and the premises of a rule we can derive its conclusion (of course without any \( \lor, \land, \exists \)-rules),
and conversely that we can derive the axioms by means of the $\lor, \land, \exists$-rules. This is left as an exercise.

The left premise in each of the elimination rules $\lor^{-}$, $\land^{-}$ and $\exists^{-}$ is called major premise (or main premise), and each of the right premises minor premise (or side premise).

4.2. Conversion. In addition to the $\rightarrow, \forall$-conversions treated in 3.1, we consider the following conversions:

$\lor$-conversion.

$$
\begin{array}{c}
\frac{M \quad [u : A] \quad [v : B]}{A} \\
\hline
\frac{A \lor B}{C} \quad \lor^{-u, v} & A \\
\hline
\frac{N \quad [K]}{C} \quad \lor^{-u, v} & A
\end{array}
$$

and

$$
\begin{array}{c}
\frac{M \quad [u : A] \quad [v : B]}{B} \\
\hline
\frac{A \lor B}{C} \quad \lor^{-u, v} & B \\
\hline
\frac{N \quad [K]}{C} \quad \lor^{-u, v} & B
\end{array}
$$

$\land$-conversion.

$$
\begin{array}{c}
\frac{M \quad N \quad [u : A] \quad [v : B]}{A} \\
\hline
\frac{A \land B}{C} \quad \land^{-u, v} & A \quad B
\end{array}
$$

$\exists$-conversion.

$$
\begin{array}{c}
\frac{M \quad [u : A]}{A[x := r]} \\
\hline
\frac{\exists x A}{\exists^{-x, u} B} \quad \exists^{-x, u} & A[x := r] \quad B \quad \exists^{-x, u} & N' \quad B
\end{array}
$$

4.3. Permutative Conversion. In a permutative conversion we permute an E-rule upwards over the minor premises of $\lor^{-}$, $\land^{-}$ or $\exists^{-}$.

$\lor$-perm conversion.

$$
\begin{array}{c}
\frac{M \quad N \quad K \quad L}{A \lor B \quad C \quad C \quad C' \quad L} \\
\hline
\frac{D \quad C' \quad \lor \quad E-rule}{C \quad C' \quad \lor \quad E-rule}
\end{array}
$$

$$
\begin{array}{c}
\frac{M \quad N \quad L \quad K \quad L}{A \lor B \quad C \quad C' \quad E-rule \quad C \quad C' \quad E-rule} \\
\hline
\frac{D \quad D \quad E-rule}{D \quad D \quad E-rule}
\end{array}
$$
\(*-perm conversion.
\[
\begin{array}{c|c|c|c}
M & N & K \\
A \land B & C & D \\
\hline
C & C' \\
\end{array}
\rightarrow
\]
E-rule

\(\exists-\)perm conversion.
\[
\begin{array}{c|c|c|c}
M & N & K \\
A \land B & C & D \\
\hline
B & C \\
\end{array}
\rightarrow
\]
E-rule

4.4. Derivations as Terms. The term representation of derivations has to be extended. The rules for \(V, \land\) and \(\exists\) with the corresponding terms are given in the table below.

The introduction rule \(\exists^+\) has as its left premise the witnessing term \(r\) to be substituted. The elimination rule \(\exists^-u\) is subject to an (Eigen-) variable condition: The derivation term \(N\) should not contain any open assumptions apart from \(u : A\) whose assumption formula contains \(x\) free, and moreover \(B\) should not contain the variable \(x\) free.

4.5. Permutative Conversions. In this section we shall write derivation terms without formula superscripts. We usually leave implicit the extra (formula) parts of derivation constants and for instance write \(\exists^+, \exists^-\) instead of \(\exists^+_{x,A}, \exists^-_{x,A,B}\). So we consider derivation terms \(M, N, K\) of the forms

\[
u | \lambda v M | \lambda y M | \lor_0 M | \lor_1 M | \langle M, N \rangle | \exists^+ r M | M N | M r | M(v_0, N_0, v_1, N_1) | M(v, w, N) | M(v, N);
\]

in these expressions the variables \(y, v, v_0, v_1, w\) get bound.

To simplify the technicalities, we restrict our treatment to the rules for \(\rightarrow\) and \(\exists\). It can easily be extended to the full set of rules; some details for disjunction are given in 4.6. So we consider

\[
u | \lambda v M | \exists^+ r M | M N | M(v, N);
\]
in these expressions the variable \(v\) gets bound.

We reserve the letters \(E, F, G\) for eliminations, i.e. expressions of the form \((v, N)\), and \(R, S, T\) for both terms and eliminations. Using this notation we obtain a second (and clearly equivalent) inductive definition of terms:

\[
u M | u \bar{M} E | \lambda v M | \exists^+ r M |
(\lambda v M) N \bar{R} | \exists^+ r M(v, N) \bar{R} | u \bar{M} E R \bar{S}.
\]
<table>
<thead>
<tr>
<th>derivation</th>
<th>term</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \frac{M}{A} ) ( \frac{A \lor B}{A \lor B} ) ( \frac{\lor_0^+}{\lor_1^+} )</td>
<td>( (\lor_{0,B}^+ M^A)^{A \lor B} ) ( (\lor_{1,A}^+ M^B)^{A \lor B} )</td>
</tr>
<tr>
<td>( \frac{[u : A]}{A \lor B} ) ( \frac{[v : B]}{A \lor B} ) ( \frac{N \lor K}{C} )</td>
<td>( (M^{A \lor B}(u^A \cdot N^C, v^B \cdot K^C))^C )</td>
</tr>
<tr>
<td>( \frac{M}{A} ) ( \frac{N}{B} ) ( \frac{A \land B}{A \land B} ) ( \land^+ )</td>
<td>( (M^A, N^B)^{A \land B} )</td>
</tr>
<tr>
<td>( \frac{[u : A]}{A \land B} ) ( \frac{[v : B]}{A \land B} ) ( \frac{N \land C}{C} )</td>
<td>( (M^{A \land B}(u^A \cdot v^B \cdot N^C))^C )</td>
</tr>
<tr>
<td>( \frac{M}{r} ) ( \frac{A[x := r]}{\exists x A} ) ( \exists^+ )</td>
<td>( (\exists^+_{x,A} r M^{A[x := r]})^\exists A )</td>
</tr>
<tr>
<td>( \frac{[u : A]}{\exists x A} ) ( \frac{N}{B} ) ( \frac{\exists x A}{\exists x, u} ) (var.cond.)</td>
<td>( (M^{\exists x A}(u^A \cdot N^B))^B ) (var.cond.)</td>
</tr>
</tbody>
</table>

**Table 3.** Derivation terms for \( \lor, \land \) and \( \exists \)

Here the final three forms are not normal: \((\lambda v M) N \bar{R}\) and \(\exists^+ r M(v.N) \bar{R}\) both are \(\beta\)-redexes, and \(u \overrightarrow{M} \overrightarrow{E} \overrightarrow{S}\) is a permutable redex. The conversion rules are

\[
\begin{align*}
(\lambda v M) N & \rightarrow_{\beta} M[v := N] & \text{\(\beta\)-conversion,} \\
\exists^+_{x,A} r M(v.N) & \rightarrow_{\beta} N[x := r][v := M] & \text{\(\beta\)-conversion,} \\
M(v.N) R & \rightarrow_{\pi} M(v.N R) & \text{permutable conversion.}
\end{align*}
\]

The *closure* of these conversions is defined by
4. NORMALIZATION INCLUDING PERMUTATIVE CONVERSIONS

- If $M \rightarrow^f M'$ or $M \rightarrow^e M'$, then $M \rightarrow M'$.
- If $M \rightarrow M'$, then also $M \体 \rightarrow M' \体$, $NN \rightarrow NM'$, $N(v,M) \rightarrow N(v,M')$, $\lambda v M \rightarrow \lambda v M'$, $\exists^+ r M \rightarrow \exists^+ r M'$ (inner reductions).

We now give the rules to inductively generate a set $SN$:

$$
\frac{M \in SN}{\mu M \in SN} \quad (\text{Var}_0) \\
\frac{M \in SN}{\lambda v M \in SN} \quad (\lambda) \\
\frac{M \in SN}{\exists^+ r M \in SN} \quad (\exists)
$$

$$
\frac{M, N \in SN}{u\tilde{M}(v,N) \in SN} \quad (\text{Var}) \\
\frac{u\tilde{M}(v.NR)S \in SN}{u\tilde{M}(v.N)RS \in SN} \quad (\text{Var}_e)
$$

$$
\frac{M[v := N]R \in SN}{N \in SN} \quad (\beta_\rightarrow) \\
\frac{M \in SN}{M[x := r][v := M]R \in SN} \quad (\beta_\leftarrow)
$$

where in $(\text{Var}_e)$ we require that $v$ is not free in $R$.

Write $M \downarrow$ to mean that $M$ is strongly normalizable, i.e., that every reduction sequence starting from $M$ terminates. By analyzing the possible reduction steps we now show that the set $\mathcal{W} := \{ M \mid M \downarrow \}$ has the closure properties of the definition of $SN$ above, and hence $SN \subseteq \mathcal{W}$.

**Lemma.** Any term in $SN$ is strongly normalizable.

**Proof.** We distinguish cases according to the generation rule of $SN$ applied last. The following rules deserve special attention.

**Case $(\text{Var})$.** We prove, as an auxiliary lemma, that

$$
u\tilde{M}(v.NR)S \downarrow \text{ implies } u\tilde{M}(v.N)RS \downarrow,$$

by induction on $u\tilde{M}(v.NR)S \downarrow$ (i.e., on the reduction tree of this term). We consider the possible reduces of $u\tilde{M}(v.N)RS$. The only interesting case is $RS = (v'.N')T\bar{T}$ and we have a permutative conversion of $R = (v'.N')$ with $T$, leading to the term $M = u\tilde{M}(v.N)(v'.N'T)\bar{T}$. Now $M \downarrow$ follows, since

$$
u\tilde{M}(v.NR)S = u\tilde{M}(v.N(v'.N'))T\bar{T}$$

leads in two permutative steps to $u\tilde{M}(v.N(v'.N'T))\bar{T}$, hence for this term we have the induction hypothesis available.

**Case $(\beta_\rightarrow)$.** We show that $M[v := N]R \downarrow$ and $N \downarrow$ imply $(\lambda v M)N R \downarrow$. This is done by a induction on $N \downarrow$, with a side induction on $M[v := N]R \downarrow$. We need to consider all possible reduces of $(\lambda v M)N R$. In case of an outer $\beta$-reduction use the assumption. If $N$ is reduced, use the induction hypothesis. Reductions in $M$ and in $R$ as well as permutative reductions within $R$ are taken care of by the side induction hypothesis.

**Case $(\beta_\leftarrow)$.** We show that $N[x := r][v := M]R \downarrow$ and $M \downarrow$ together imply $\exists^+ r M(v.N) R \downarrow$. This is done by a threefold induction: first on $M \downarrow$, second on $N[x := r][v := M]R \downarrow$, and third on the length of $R$. We need to consider all possible reduces of $\exists^+ r M(v.N) R$. In case of an outer $\beta$-reduction use the
assumption. If $M$ is reduced, use the first induction hypothesis. Reductions in $N$ and in $R$ as well as permutative reductions within $R$ are taken care of by the second induction hypothesis. The only remaining case is when $R = SS\tilde{F}$ and $(v.N)\tilde{S}$ is permuted with $S$, to yield $\exists^+ r M(v.NS)\tilde{S}$. Apply the third induction hypothesis, since $(NS)[x := r][v := M]\tilde{S} = N[x := r][v := M]\tilde{S}$. 

For later use we prove a slightly generalized form of the rule $(\text{Var}_\pi)$:

**Proposition.** If $M(v.NR)\tilde{S} \in SN$, then $M(v.N)RS\tilde{S} \in SN$.

**Proof.** Induction on the generation of $M(v.NR)\tilde{S} \in SN$. We distinguish cases according to the form of $M$.

**Case** $u\tilde{T}(v.NR)\tilde{S} \in SN$. If $\tilde{T} = \tilde{M}$, use $(\text{Var}_\pi)$. Otherwise we have $u\tilde{M}(v',N')\tilde{R}(v.NR)\tilde{S} \in SN$. This must be generated by repeated applications of $(\text{Var}_\pi)$ from $u\tilde{M}(v',N')R(v.NR)\tilde{S} \in SN$, and finally by $(\text{Var})$ from $M \in SN$ and $N'\tilde{R}(v.NR)\tilde{S} \in SN$. The induction hypothesis for the latter yields $N'\tilde{R}(v.N)RS\tilde{S} \in SN$, hence $u\tilde{M}(v'.N')R(v.N)RS\tilde{S} \in SN$ by $(\text{Var})$ and finally $u\tilde{M}(v'.N')R(v.N)RS\tilde{S} \in SN$ by $(\text{Var}_\pi)$.

**Case** $\exists^+ r M\tilde{T}(v.NR)\tilde{S} \in SN$. Similarly, with $(\beta_3)$ instead of $(\text{Var}_\pi)$. In detail: If $\tilde{T}$ is empty, by $(\beta_3)$ this came from $(NR)[x := r][v := M]\tilde{S} = N[x := r][v := M]RS\tilde{S} \in SN$ and $M \in SN$, hence $\exists^+ r M(v.N)RS\tilde{S} \in SN$ again by $(\beta_3)$. Otherwise we have $\exists^+ r M(v'.N')\tilde{T}(v.NR)\tilde{S} \in SN$. This must be generated by $(\beta_3)$ from $N'[x := r][v' := M]\tilde{T}(v.NR)\tilde{S} \in SN$. The induction hypothesis yields $N'[x := r][v' := M]\tilde{T}(v.N)RS\tilde{S} \in SN$, hence $\exists^+ r M(v'.N')\tilde{T}(v.N)RS\tilde{S} \in SN$ by $(\beta_3)$.

**Case** $(\lambda v M)N'\tilde{R}(w.NR)\tilde{S} \in SN$. By $(\beta_\to)$ this came from $N' \in SN$ and $M[v := N']\tilde{R}(w.NR)\tilde{S} \in SN$. The induction hypothesis yields $M[v := N']\tilde{R}(w.N)RS\tilde{S} \in SN$, hence $(\lambda v M)N'\tilde{R}(w.N)RS\tilde{S} \in SN$ by $(\beta_\to)$. 

In what follows we shall show that every term is in SN and hence is strongly normalizable. Given the definition of SN we only have to show that SN is closed under $\to\tilde{\gamma}$ and $\exists\tilde{\gamma}$. In order to prove this we must prove simultaneously the closure of SN under substitution.

**Theorem (Properties of SN).** For all formulas $A$,

(a) for all $M \in SN$, if $M$ proves $A = A_0 \to A_1$ and $N \in SN$, then $MN \in SN$,
(b) for all $M \in SN$, if $M$ proves $A = \exists x B$ and $N \in SN$, then $M(v.N) \in SN$,
(c) for all $M \in SN$, if $N^A \in SN$, then $M[v := N] \in SN$.

**Proof.** Induction on $\phi(A)$. We prove (a) and (b) before (c), and hence have (a) and (b) available for the proof of (c). More formally, by induction on $A$ we simultaneously prove that (a) holds, that (b) holds and that (a), (b) together imply (c).

(a). By induction on $M \in SN$. Let $M \in SN$ and assume that $M$ proves $A = A_0 \to A_1$ and $N \in SN$. We distinguish cases according to how $M \in SN$ was generated. For $(\text{Var}_0)$, $(\text{Var}_{\pi})$, $(\beta_{\to})$ and $(\beta_3)$ use the same rule again.

**Case** $u\tilde{M}(v.N') \in SN$ by $(\text{Var})$ from $\tilde{M}, N' \in SN$. Then $N'N \in SN$ by side induction hypothesis for $N'$, hence $u\tilde{M}(v.N')N \in SN$ by $(\text{Var})$, hence $u\tilde{M}(v.N')N \in SN$ by $(\text{Var}_{\pi})$. 

Case \((\lambda v M)A_0 \rightarrow A_1 \in \text{SN}\) by (\(\lambda\)) from \(M \in \text{SN}\). Use (\(\beta_{\rightarrow}\)); for this we need to know \(M[v := N] \in \text{SN}\). But this follows from \(\text{III}(c)\) for \(M\), since \(N\) derives \(A_0\).

(b). By induction on \(M \in \text{SN}\). Let \(M \in \text{SN}\) and assume that \(M\) proves \(A = \exists x B\) and \(N \in \text{SN}\). The goal is \(M(v,N) \in \text{SN}\). We distinguish cases according to how \(M \in \text{SN}\) was generated. For \((\text{Var}_v)\), \((\beta_{\rightarrow})\) and \((\beta_{\exists})\) use the same rule again.

Case \(u\vec{M} \in \text{SN}\) by \((\text{Var}_0)\) from \(\vec{M} \in \text{SN}\). Use \((\text{Var})\).

Case \((\exists + r) M \exists x A \in \text{SN}\) by \((\exists)\) from \(M \in \text{SN}\). Use \((\beta_{\exists})\); for this we need to know \(N[x := \tau][v := M] \in \text{SN}\). But this follows from \(\text{III}(c)\) for \(N[x := \tau]\), since \(M\) derives \(A[x := \tau]\).

Case \(u\vec{M}(v',N') \in \text{SN}\) by \((\text{Var})\) from \(\vec{M}, N' \in \text{SN}\). Then \(N'(v,N) \in \text{SN}\) by side induction hypothesis for \(N'\), hence \(u\vec{M}(v,N'(v,N)) \in \text{SN}\) by \((\text{Var})\) and therefore \(u\vec{M}(v,N')(v,N) \in \text{SN}\) by \((\text{Var}_v)\).

(c). By induction on \(M \in \text{SN}\). Let \(N^A \in \text{SN}\); the goal is \(M[v := N] \in \text{SN}\). We distinguish cases according to how \(M \in \text{SN}\) was generated. For \((\lambda)\), \((\exists)\), \((\beta_{\rightarrow})\) and \((\beta_{\exists})\) use the same rule again.

Case \(u\vec{M} \in \text{SN}\) by \((\text{Var}_0)\) from \(\vec{M} \in \text{SN}\). Then \(\vec{M}[v := N] \in \text{SN}\) by \(\text{III}(c)\). If \(u \neq v\), use \((\text{Var}_0)\) again. If \(u = v\), we must show \(N\vec{M}[v := N] \in \text{SN}\). Note that \(N\) proves \(A\); hence the claim follows from (a) and the induction hypothesis.

Case \(u\vec{M}(v',N') \in \text{SN}\) by \((\text{Var})\) from \(\vec{M}, N' \in \text{SN}\). If \(u \neq v\), use \((\text{Var})\) again. If \(u = v\), we must show \(N\vec{M}[v := N](v',N'[v := N]) \in \text{SN}\). Note that \(N\) proves \(A\); hence in case \(\vec{M}\) empty the claim follows from (b), and otherwise from (a) and the induction hypothesis.

Case \(u\vec{M}(v',N')R\vec{S} \in \text{SN}\) by \((\text{Var}_v)\) from \(u\vec{M}(v',N'R)\vec{S} \in \text{SN}\). If \(u \neq v\), use \((\text{Var}_v)\) again. If \(u = v\), from the induction hypothesis we obtain
\[
N\vec{M}[v := N](v',N'[v := N]R[v := N]).\vec{S}[v := N] \in \text{SN}
\]
Now use the proposition above. \(\square\)

**COROLLARY.** Every term is strongly normalizable.

**PROOF.** Induction on the (first) inductive definition of terms \(M\). In cases \(u\) and \(\lambda v M\) the claim follows from the definition of \(\text{SN}\), and in cases \(MN\) and \(M(v,N)\) it follows from parts (a), (b) of the previous theorem. \(\square\)

### 4.6. Disjunction.
We describe the changes necessary to extend the result above to the language with disjunction \(\lor\).

We have additional \(\beta\)-conversions
\[
\bigvee_i^+ M(v_0,N_0,v_1.N_1) \rightarrow_\beta M[v_i := N_i] \quad \beta_{\lor,i}\text{-conversion.}
\]

The definition of \(\text{SN}\) needs to be extended by
\[
\frac{M \in \text{SN}}{\bigvee_i^+ M \in \text{SN}} (\lor_i)
\]
\[
\frac{\vec{M},N_0,N_1 \in \text{SN} \quad (\text{Var}_v) \quad u\vec{M}(v_0.N_0.R,v_1.N_1.R)\vec{S} \in \text{SN}}{u\vec{M}(v_0.N_0,v_1.N_1) \in \text{SN}} (\text{Var}_{\lor,v})
\]
\[ N_i[v_i := M]\overline{R} \in SN \quad N_{1-i}[\overline{R}] \in SN \quad M \in SN \]

\[ \forall^+ i M(v_0.N_0, v_1.N_1)\overline{R} \in SN \quad (\beta_{\forall i}) \]

The former rules (\(\var\)), (\(\var_{\eta}\)) should then be renamed into (\(\var_{\exists}\)), (\(\var_{\exists, \eta}\)).

The lemma above stating that every term in SN is strongly normalizable needs to be extended by an additional clause:

\textbf{Case} (\(\beta_{\forall i}\)). We show that \(N_i[v_i := M]\overline{R} \downarrow\), \(N_{1-i}[\overline{R}] \downarrow\) and \(M \downarrow\) together imply \(\forall^+ i M(v_0.N_0, v_1.N_1)\overline{R} \downarrow\). This is done by a fourfold induction: first on \(M \downarrow\), second on \(N_i[v_i := M]\overline{R} \downarrow\), \(N_{1-i}[\overline{R}] \downarrow\), third on \(N_{1-i}[\overline{R}] \downarrow\) and fourth on the length of \(\overline{R}\). We need to consider all possible reducts of \(\forall^+ i M(v_0.N_0, v_1.N_1)\overline{R}\). In case of an outer \(\beta\)-reduction use the assumption. If \(M\) is reduced, use the first induction hypothesis. Reductions in \(N_i\) and in \(\overline{R}\) as well as permutative reductions within \(\overline{R}\) are taken care of by the second induction hypothesis. Reductions in \(N_{1-i}\) are taken care of by the third induction hypothesis. The only remaining case is when \(\overline{R} = SS\overline{R}\) and \((v_0.N_0, v_1.N_1)\) is permuted with \(S\), to yield \((v_0.N_0 S, v_1.N_1 S)\). Apply the fourth induction hypothesis, since \((N_i S)[v := M]\overline{S} = N_i[v := M]S \overline{S}\).

Finally the theorem above stating properties of SN needs an additional clause:

- for all \(M \in SN\), if \(M\) proves \(A = A_0 \lor A_1\) and \(N_0, N_1 \in SN\), then \(M(v_0.N_0, v_1.N_1) \in SN\).

\textbf{Proof.} The new clause is proved by induction on \(M \in SN\). Let \(M \in SN\) and assume that \(M\) proves \(A = A_0 \lor A_1\) and \(N_0, N_1 \in SN\). The goal is \(M(v_0.N_0, v_1.N_1) \in SN\). We distinguish cases according to how \(M\) was generated. For \((\var_{\exists, \eta})\), \((\var_{\forall, \eta})\), \((\beta_{\exists})\) and \((\beta_{\forall i})\) use the same rule again.

\textbf{Case} \(u \overline{M} \in SN\) by \((\var_0)\) from \(\overline{M} \in SN\). Use \((\var_{\forall})\).

\textbf{Case} \((\forall^+ i M)^{A_0 \lor A_1} \in SN\) by \((\forall_i)\) from \(M \in SN\). Use \((\beta_{\forall i})\); for this we need to know \(N_i[v_i := M] \in SN\) and \(N_{1-i} \in SN\). The latter is assumed, and the former follows from main induction hypothesis (with \(N_i\)) for the substitution clause of the theorem, since \(M\) derives \(A_i\).

\textbf{Case} \(u \overline{M}(v'.N') \in SN\) by \((\var_{\exists})\) from \(\overline{M}, N' \in SN\). For brevity let \(E := (v_0.N_0, v_1.N_1)\). Then \(N'E \in SN\) by side induction hypothesis for \(N'\), so \(u \overline{M}(v'.N'E) \in SN\) by \((\var_{\exists})\) and therefore \(u \overline{M}(v'.N')E \in SN\) by \((\var_{\exists, \eta})\).

\textbf{Case} \(u \overline{M}(v_0.N'_0, v_1.N'_1) \in SN\) by \((\var_{\forall})\) from \(\overline{M}, N'_0, N'_1 \in SN\). Let \(E := (v_0.N_0, v_1.N_1)\). Then \(N'E \in SN\) by side induction hypothesis for \(N'_1\), so \(u \overline{M}(v_0.N_0 E, v_1.N'_1 E) \in SN\) by \((\var_{\forall})\) and therefore \(u \overline{M}(v_0.N'_0, v_1.N'_1)E \in SN\) by \((\var_{\forall, \eta})\).

Clause (c) now needs additional cases, e.g.,

\textbf{Case} \(u \overline{M}(v_0.N_0, v_1.N_1) \in SN\) by \((\var_{\forall})\) from \(\overline{M}, N_0, N_1 \in SN\). If \(u \neq v\), use \((\var_{\forall})\). If \(u = v\), we show \(N \overline{M}[v := N](v_0.N_0[v := N], v_1.N_1[v := N]) \in SN\). Note that \(N\) proves \(A_i\); hence in case \(\overline{M}\) empty the claim follows from (b), and otherwise from (a) and the induction hypothesis.

\[ \square \]

\textbf{4.7. The Structure of Normal Derivations.} As mentioned already, normalizations aim at removing local maxima of complexity, i.e. formula occurrences which are first introduced and immediately afterwards eliminated.
However, an introduced formula may be used as a minor premise of an application of \(\lor^-, \land^-\) or \(\exists^-\), then stay the same throughout a sequence of applications of these rules, being eliminated at the end. This also constitutes a local maximum, which we should like to eliminate; for that we need the so-called permutative conversions. First we give a precise definition.

**Definition.** A *segment* of (length \(n\)) in a derivation \(M\) is a sequence \(A_1, \ldots, A_n\) of occurrences of a formula \(A\) such that

(a) for \(1 < i < n\), \(A_i\) is a minor premise of an application of \(\lor^-, \land^-\) or \(\exists^-\), with conclusion \(A_{i+1}\);
(b) \(A_n\) is not a minor premise of \(\lor^-, \land^-\) or \(\exists^-\).
(c) \(A_1\) is not the conclusion of \(\lor^-, \land^-\) or \(\exists^-\).

(Note: An f.o. which is neither a minor premise nor the conclusion of an application of \(\lor^-, \land^-\) or \(\exists^-\) always belongs to a segment of length 1.) A segment is *maximal* or a *cut (segment)* if \(A_n\) is the major premise of an E-rule, and either \(n > 1\), or \(n = 1\) and \(A_1 = A_n\) is the conclusion of an I-rule.

We shall use \(\sigma, \sigma'\) for segments. We shall say that \(\sigma\) is a *subformula* of \(\sigma'\) if the formula \(A\) in \(\sigma\) is a subformula of \(B\) in \(\sigma'\). Clearly a derivation is normal iff it does not contain a maximal segment.

The argument in 3.7 needs to be refined to also cover the rules for \(\lor, \land, \exists\). The reason for the difficulty is that in the E-rules \(\lor^-, \land^-, \exists^-\) the subformulas of a major premise \(A \lor B\), \(A \land B\) or \(\exists x A\) of an E-rule application do not appear in the conclusion, but among the assumptions being discharged by the application. This suggests the definition of a track below.

The general notion of a track is designed to retain the subformula property in case one passes through the major premise of an application of a \(\lor^-, \land^-, \exists^-\)-rule. In a track, when arriving at an \(A_i\) which is the major premise of an application of such a rule, we take for \(A_{i+1}\) a hypothesis discharged by this rule.

**Definition.** A *track* of a derivation \(M\) is a sequence of f.o.’s \(A_0, \ldots, A_n\) such that

(a) \(A_0\) is a top f.o. in \(M\) not discharged by an application of an \(\lor^-, \land^-, \exists^-\)-rule;
(b) \(A_i\) for \(i < n\) is not the minor premise of an instance of \(\rightarrow^-, \land^-\), and either
   (i) \(A_i\) is not the major premise of an instance of a \(\lor^-, \land^-, \exists^-\)-rule and \(A_{i+1}\) is directly below \(A_i\), or
   (ii) \(A_i\) is the major premise of an instance of a \(\lor^-, \land^-, \exists^-\)-rule and \(A_{i+1}\) is an assumption discharged by this instance;
(c) \(A_n\) is either
   (i) the minor premise of an instance of \(\rightarrow^-, \land^-\), or
   (ii) the conclusion of \(M\), or
   (iii) the major premise of an instance of a \(\lor^-, \land^-, \exists^-\)-rule in case there are no assumptions discharged by this instance.

**Proposition.** Let \(M\) be a normal derivation, and let \(\pi = \sigma_0, \ldots, \sigma_n\) be a track in \(M\). Then there is a segment \(\sigma_i\) in \(\pi\), the minimum segment or minimum part of the track, which separates two (possibly empty) parts of \(\pi\),...
called the E-part (elimination part) and the I-part (introduction part) of $\pi$
such that

(a) for each $\sigma_j$ in the E-part one has $j < i$, $\sigma_j$ is a major premise of an
E-rule, and $\sigma_{j+1}$ is a strictly positive part of $\sigma_j$, and therefore each $\sigma_j$
is a s.p.p. of $\sigma_0$;

(b) for each $\sigma_j$ which is the minimum segment or is in the I-part one has
$i \leq j$, and if $j \neq n$, then $\sigma_j$ is a premise of an I-rule and a s.p.p. of
$\sigma_{j+1}$, so each $\sigma_j$ is a s.p.p. of $\sigma_n$.

**Definition.** A track of order 0, or main track, in a normal derivation
is a track ending either in the conclusion of the whole derivation or in the
major premise of an application of a $\lor^-, \land^-$ or $\exists^-$-rule, provided there are
no assumption variables discharged by the application. A track of order
$n+1$ is a track ending in the minor premise of an $\to^-$-application, with
major premise belonging to a track of order $n$.

A main branch of a derivation is a branch $\pi$ in the proof tree such that $\pi$
passes only through premises of I-rules and major premises of E-rules, and
$\pi$ begins at a top node and ends in the conclusion.

**Remark.** By an obvious simplification conversion we may remove every
application of an $\lor^-, \land^-$ or $\exists^-$-rule that discharges no assumption variables.
If such simplification conversion are performed, each track of order 0 in a
normal derivation is a track ending in the conclusion of the whole derivation.

If we search for a main branch going upwards from the conclusion, the
branch to be followed is unique as long as we do not encounter an $\land^+$-
application.

**Lemma.** In a normal derivation each formula occurrence belongs to some
track.

**Proof.** By induction on the height of normal derivations. For example,
suppose a derivation $K$ ends with an $\exists^-$-application:

\[
\begin{array}{l}
[u: A] \\
M \ | \ N \\
\exists x A \ | \ B \\
\hline
B \\
\exists^- x, u
\end{array}
\]

$B$ in $N$ belongs to a track $\pi$ (induction hypothesis); either this does not
start in $u: A$, and then $\pi, B$ is a track in $K$ which ends in the conclusion; or
$\pi$ starts in $u: A$, and then there is a track $\pi'$ in $M$ (induction hypothesis)
such that $\pi', \pi, C$ is a track in $K$ ending in the conclusion. The other cases
are left to the reader. 

**Theorem (Subformula property).** Let $M$ be a normal derivation where
every application of an $\lor^-, \land^-$ or $\exists^-$-rule discharges at least one assumption
variable. Then each formula occurring in the derivation is a subformula of
either the end formula or else an assumption formula.

**Proof.** As note above, each track of order 0 in $M$ is a track ending in
the conclusion of $M$. We can now prove the theorem for tracks of order $n$,
by induction on $n$. 

THEOREM (Disjunction property). If $\Gamma$ does not contain a disjunction as s.p.p. (= strictly positive part, defined in 1.3), then, if $\Gamma \vdash A \lor B$, it follows that $\Gamma \vdash A$ or $\Gamma \vdash B$.

PROOF. Consider a normal derivation $M$ of $A \lor B$ from assumptions $\Gamma$ not containing a disjunction as s.p.p. The conclusion $A \lor B$ is the final formula of a (main) track, whose top formula $A_0$ in $M$ must be an assumption in $\Gamma$. Since $\Gamma$ does not contain a disjunction as s.p.p., the segment $\sigma$ with the conclusion $A \lor B$ is in the I-part. Skip the final $\lor^+$-rule and replace the formulas in $\sigma$ by $A$ if $i = 0$, and by $B$ if $i = 1$.

There is a similar theorem for the existential quantifier:

THEOREM (Explicit definability under hypotheses). Let $\Gamma \vdash \exists x A$.

(a) If $\Gamma$ does not contain an existential s.p.p., then there are terms $r_1$, $r_2$, \ldots, $r_n$ such that $\Gamma \vdash A[x := r_1] \lor \ldots \lor A[x := r_n]$.

(b) If $\Gamma$ neither contains a disjunctive s.p.p., nor an existential s.p.p., then there is a term $r$ such that $\Gamma \vdash A[x := r]$.

PROOF. Consider a normal derivation $M$ of $\exists x A$ from assumptions $\Gamma$ not containing an existential s.p.p. We use induction on the derivation, and distinguish cases on the last rule.

(a). By assumption the last rule cannot be $\exists^−$. We only consider the case $\lor^−$ and leave the others to the reader.

\[
\begin{array}{ccc}
[u : B] & [v : C] & \mid M \\
B \lor C & \mid N_0 & \mid N_1 \\
\exists x A & \mid N_1 & \exists x A \\
\forall u, v & \\end{array}
\]

By assumption again neither $B$ nor $C$ can have an existential s.p.p. Applying the induction hypothesis to $N_0$ and $N_1$ we obtain

\[
\begin{array}{ccc}
[u : B] & [v : C] & \mid M \\
B \lor C & \mid N_0 & \mid N_1 \\
\forall_{i=1}^n A[x := r_i] & \mid \forall_{i=1}^{n+m} A[x := r_i] & \forall_{i=1}^{n+m} A[x := r_i] \\
\forall_{i=1}^{n+m} A[x := r_i] & \forall_{i=1}^{n+m} A[x := r_i] & \forall_{i=1}^{n+m} A[x := r_i] \\
\end{array}
\]

(b). Similarly; by assumption the last rule can be neither $\lor^−$ nor $\exists^−$.

REMARK. Rasiowa-Harrop formulas (in the literature also called Harrop formulas) are formulas for which no s.p.p. is a disjunction or an existential formula. For $\Gamma$ consisting of Rasiowa-Harrop formulas both theorems above hold.

5. Notes

The proof of the existence of normal forms w.r.t. permutative conversions is originally due to Prawitz [14]. We have adapted a method developed by Joachimski and Matthes [8], which in turn is based on van Raamsdonk's and Severi's [18].