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 (1934). 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 beween existential proofs that actually construct witnessing objects, and others that don't.

An essential point for Mathematical Logic is to fix a formal language to be used. We take implication $\to$ and the universal quantifier $\forall$ as basic. Then the logic rules correspond to lambda calculus. The additional connectives are defined via axiom schemes: the existential quantifier $\exists$, disjunction $\lor$ and conjunction $\land$.

1.1. Formal Languages

1.1.1. Examples. Before describing terms and formulas of a general formal language we consider some examples. The first one is a simple arithmetical theory Q due originally to Robinson. Its language is given by the constant 0, the function symbols S, $+$, $\cdot$ and the relation symbol $=$. Its axioms are (disregarding the standard certain equality axioms)

$$\mathrm{S}x = 0 \to \bot,$$
$$\mathrm{S}x = \mathrm{S}y \to x = y,$$

$$
\begin{aligned}
&x + 0 = x,\\
&x + \mathrm{S}y = \mathrm{S}(x + y),\\
&x \cdot 0 = 0,\\
&x \cdot \mathrm{S}y = x \cdot y + x,\\
&\exists_z(x + \mathrm{S}z = y) \vee x = y \vee \exists_z(y + \mathrm{S}z = x).
\end{aligned}
$$

Here we tacitly assume that the free variables are universally quantified in front of the formula.

Another example is the theory of partial orderings. Its language is determined by just one relation symbol $\leq$, and its axioms are reflexivity and transitivity:

$$
\begin{aligned}
&x \leq x,\\
&x \leq y \wedge y \leq z \to x \leq z.
\end{aligned}
$$

We now define generally what the terms and formulas of a formal language are.

1.1.2. 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* (or *predicate symbols*). 0-ary relation symbols are called *propositional symbols*. $\bot$ (read "falsum") is required as a fixed propositional symbol. The language will *not*, unless stated otherwise, contain $=$ as a primitive. Binary relation symbols can be marked as *infix*.
- For every natural number $n \geq 0$ a (possible empty) set of n-ary *function symbols*. 0-ary function symbols are called *constants*. Binary function sysmbols can also be marked as infix.

We assume that all these sets of variables, relation and function symbols are disjoint. $\mathcal{L}$ is kept fixed and will only be mentioned when necessary.

Terms are inductively defined as follows.

- Every variable is a term.
- Every constant is a term.
- If $t_1, \dots, t_n$ are terms and f is an n-ary function symbol with $n \geq 1$, then $f(t_1, \dots, t_n)$ is a term. If r, s are terms and $\circ$ is a binary function symbol, then $(r \circ s)$ is a term.

From terms one constructs *prime formulas*, also called *atomic formulas*: If $t_1, \dots, t_n$ are terms and R is an n-ary relation symbol, then $R(t_1, \dots, t_n)$ is a prime formula. If r, s are terms and $\sim$ is a binary relation symbol, then $(r \sim s)$ is a prime formula.

Formulas are inductively defined from prime formulas by

- Every prime formula is a formula.
- If A and B are formulas, then so are $(A \to B)$ ("if A, then B"), $(A \wedge B)$ ("A and B") and $(A \vee B)$ ("A or B").
- If A is a 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 formulas.

Negation is defined by

$$\neg A := (A \to \bot).$$

NOTATION. For readability we write $\forall_x A$ for $\forall x A$, and similarly $\exists_x A$ for $\exists x A$. In writing formulas we save on parentheses by assuming that $\forall, \exists, \neg$ bind more strongly than $\wedge, \vee$, and that in turn $\wedge, \vee$ bind more strongly than $\to, \leftrightarrow$ (where $A \leftrightarrow B$ abbreviates $(A \to B) \wedge (B \to A)$). Outermost parentheses can be dropped. Thus $A \wedge \neg B \to C$ is read as $((A \wedge (\neg B)) \to C)$. In the case of iterated implications we use the short notation

$$A_1 \to A_2 \to \dots A_{n-1} \to A_n \quad \text{for} \quad A_1 \to (A_2 \to \dots (A_{n-1} \to A_n) \dots).$$

We also save on parentheses by writing for instance $Rxyz$, $Rt_0t_1t_2$ 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 fx 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 $Rfxgyhz$.

Weak disjunction and the weak existential quantifier are defined by

$$\begin{aligned} A \mathbin{\tilde{\vee}} B &:= \neg A \to \neg B \to \bot, \\ \tilde{\exists}_x A \quad &:= \neg \forall_x \neg A. \end{aligned}$$

The connectives $\tilde{\vee}$, $\tilde{\exists}$ are also called "classical", because this is what remains when one disregards constructive content, as is usually done in present-day mathematics.

1.1.3. 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 α-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 name-free terms of de Bruijn (1972). 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 $\mathrm{FV}(t)$ is the set of variables free in the term t, $\mathrm{FV}(A)$ 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 $\vec{t} = t_1, \dots, t_n$ for the variables $\vec{x} = x_1, \dots, x_n$, 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. □

1.1.4. 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 $\circ = \to, \wedge, \vee$;
(c) if $\forall_x B(x)$ or $\exists_x B(x)$ is a subformula of A, then so is $B(t)$, 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 $B \wedge C$ or $B \vee C$ is a positive (negative, strictly positive) subformula of A, then so are B, C;
(c) if $\forall_x B(x)$ or $\exists_x B(x)$ is a positive (negative, strictly positive) subformula of A, then so is $B(t)$;
(d) if $B \to C$ 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 $B \to C$ 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. $(P \to Q) \to R \wedge \forall_x S(x)$ has as s.p.p.'s the whole formula, $R \wedge \forall_x S(x)$, R, $\forall_x S(x)$, $S(t)$. The positive subformulas are the s.p.p.'s and in addition P; the negative subformulas are $P \to Q$, Q.

1.2. 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 $\to$ 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 $\wedge$, disjunction $\vee$ and the existential quantifier $\exists$. The resulting system is called *minimal logic*; it has been introduced by Kolmogorov (1925) and Johansson (1937). 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 embed classical logic, we add as an axiom scheme the principle of *indirect proof*, also called *stability*. However, then it is appropriate to restrict to the language based on $\to$, $\forall$, $\bot$ and $\wedge$; we can introduce weak disjunction $\tilde{\vee}$ and the weak existential quantifier $\tilde{\exists}$ via their definitions above. For these the usual introduction and elimination properties can then be derived.

1.2.1. Examples of derivations.

1.2.1. 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.

$$(A \to B \to C) \to (A \to B) \to A \to C.$$

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

$$\forall_x (A \to B) \to A \to \forall_x B, \quad \text{if } x \notin \mathrm{FV}(A).$$

Informal proof. Assume $\forall_x(A \to B)$. To show: $A \to \forall_x B$. 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 $A \to B$, by the first assumption. Hence also B, by the second assumption. □

$$(A \to \forall_x B) \to \forall_x(A \to B), \quad \text{if } x \notin \mathrm{FV}(A).$$

Informal proof. Assume $A \to \forall_x B$. To show: $\forall_x(A \to B)$. Let x be arbitrary; note that we have not made any assumptions on x. To show: $A \to B$. 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 ν are the *premises* of the rule application, the formula at node ν 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* $\Box_0, \Box_1, \dots$, denoted by $u, v, w, u_0, u_1, \dots$. 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 more than once (this was the case in the first example), the assumption marked with u (written $u\colon A$) 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 $u\colon A$ marked with that u). Then bound assumption variable is added to the label of the node. An application of the $\forall$-introduction rule similarly binds an object variable x (and hence removes the dependency on x).

1.2.2. Introduction and elimination rules for $\to$ and $\forall$. We now formulate the rules of natural deduction. First we have an assumption rule, allowing to write down an arbitrary formula A together with a marker u:

$$u\colon A \quad \text{Assumption}$$

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

$$\dfrac{\begin{array}{c}[u\colon A]\\ \mid M\\ B\end{array}}{A \to B}\,\to^+ u \qquad \dfrac{\begin{array}{c}\mid M\\ A \to B\end{array} \quad \begin{array}{c}\mid N\\ A\end{array}}{B}\,\to^-$$

For the universal quantifier $\forall$ there is an introduction rule $\forall^+$ and an elimination rule $\forall^-$, whose right premise is the term r to be substituted. The rule $\forall^+$ with conclusion $\forall_x A$ 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.

$$\dfrac{\begin{array}{c}\mid M\\ A\end{array}}{\forall_x A}\,\forall^+ \qquad \dfrac{\begin{array}{c}\mid M\\ \forall_x A(x)\end{array} \quad r}{A(r)}\,\forall^-$$

We now give derivations for the example formulas above. 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,

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{u\colon A \to B \to C \quad w\colon A}{B \to C} \quad \dfrac{v\colon A \to B \quad w\colon A}{B}}{C}}{A \to C}\,\to^+ w}{(A \to B) \to A \to C}\,\to^+ v}{(A \to B \to C) \to (A \to B) \to A \to C}\,\to^+ u$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{u\colon \forall_x(A \to B) \quad x}{A \to B} \quad v\colon A}{B}}{\forall_x B}\,\forall^+}{A \to \forall_x B}\,\to^+ v}{\forall_x(A \to B) \to A \to \forall_x B}\,\to^+ u$$

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

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{u\colon A \to \forall_x B \qquad v\colon A}{\forall_x B} \qquad x}{B}}{A \to B}\,\to^+ v}{\forall_x(A \to B)}\,\forall^+}{(A \to \forall_x B) \to \forall_x(A \to B)}\,\to^+ u$$

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

Recall that negation is defined by $\neg A := (A \to \bot)$. The following can easily be derived.

$$A \to \neg\neg A,$$
$$\neg\neg\neg A \to \neg A.$$

However, $\neg\neg A \to A$ is in general *not* derivable with our (minimal) logic rules. We will come back to this question later on.

Lemma. *The following are derivable.*

$$(A \to B) \to \neg B \to \neg A,$$
$$\neg(A \to B) \to \neg B,$$
$$\neg\neg(A \to B) \to \neg\neg A \to \neg\neg B,$$
$$(\bot \to B) \to (\neg\neg A \to \neg\neg B) \to \neg\neg(A \to B),$$
$$\neg\neg\forall_x A \to \forall_x\neg\neg A.$$

The proof is left as an exercise.

1.2.3. Disjunction $\vee$, conjunction $\wedge$ and existence $\exists$. We introduce the remaining logical connectives by axiom schemes. They come in two forms: for each of $\vee, \wedge, \exists$ we have introduction axioms and one elimination axiom. In writing these axioms we follow the usual practice of considering all free variables in an axiom as universally quantified outside.

For disjunction the introduction and elimination axioms are

$$\vee_0^+\colon A \to A \vee B,$$
$$\vee_1^+\colon B \to A \vee B,$$
$$\vee^-\colon A \vee B \to (A \to C) \to (B \to C) \to C.$$

For conjunction we have

$$\wedge^+\colon A \to B \to A \wedge B, \qquad \wedge^-\colon A \wedge B \to (A \to B \to C) \to C$$

and for the existential quantifier

$$\exists^+ \colon A \to \exists_x A, \qquad \exists^- \colon \exists_x A \to \forall_x (A \to B) \to B \quad (x \notin \mathrm{FV}(B)).$$

REMARK. All these axioms can be seen as special cases of a general scheme, that of an *inductively defined predicate*, which is defined by some introduction rules and one elimination rule. Later we will study this kind of definition in full generality. The desire to follow this general pattern is also the reason that we have chosen our rather strange $\wedge^-$-axiom, instead of the more usual $A \wedge B \to A$ and $A \wedge B \to B$ (which clearly are equivalent).

DEFINITION. A formula A is called *derivable* (in *minimal logic*), written $\vdash A$, if there is a derivation of A (without free assumptions) from the axioms above using the natural deduction rules in 1.2.2. A formula B is called derivable from assumptions $A_1, \dots, A_n$, if there is a derivation of B with free assumptions among $A_1, \dots, A_n$. Let Γ be a (finite or infinite) set of formulas. We write $\Gamma \vdash B$ if the formula B is derivable from finitely many assumptions $A_1, \dots, A_n \in \Gamma$.

In practice it will be convenient to use *rules* rather than axioms. Again for each of $\vee, \wedge, \exists$ we have introduction and elimination rules (E-rules). For disjunction the introduction and elimination rules are

$$\frac{\begin{array}{c}\mid M\\ A\end{array}}{A \vee B}\,\vee_0^+ \qquad \frac{\begin{array}{c}\mid M\\ B\end{array}}{A \vee B}\,\vee_1^+ \qquad \frac{\begin{array}{c}\\ \mid M\\ A \vee B\end{array} \quad \begin{array}{c}[u\colon A]\\ \mid N\\ C\end{array} \quad \begin{array}{c}[v\colon B]\\ \mid K\\ C\end{array}}{C}\,\vee^- u, v$$

For conjunction we have

$$\frac{\begin{array}{c}\mid M\\ A\end{array} \quad \begin{array}{c}\mid N\\ B\end{array}}{A \wedge B}\,\wedge^+ \qquad \frac{\begin{array}{c}\\ \mid M\\ A \wedge B\end{array} \quad \begin{array}{c}[u\colon A] \quad [v\colon B]\\ \mid N\\ C\end{array}}{C}\,\wedge^- u, v$$

and for the existential quantifier

$$\frac{r \quad \begin{array}{c}\mid M\\ A(r)\end{array}}{\exists_x A(x)}\,\exists^+ \qquad \frac{\begin{array}{c}\\ \mid M\\ \exists_x A\end{array} \quad \begin{array}{c}[u\colon A]\\ \mid N\\ B\end{array}}{B}\,\exists^- x, u \text{ (var.cond.)}$$

The rule $\exists^- x, u$ is subject to the following *(Eigen-)variable condition*: The derivation N should not contain any open assumptions apart from $u\colon A$ 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 $\vee$, $\wedge$, $\exists$ 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 $\vee, \wedge, \exists$-rules), and conversely that we can derive the axioms by means of the $\vee, \wedge, \exists$-rules. This is left as an exercise.

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

We collect some easy facts about derivability; $B \leftarrow A$ means $A \to B$.

Lemma. *The following are derivable.*

$$(A \wedge B \to C) \leftrightarrow (A \to B \to C),$$
$$(A \to B \wedge C) \leftrightarrow (A \to B) \wedge (A \to C),$$

$$(A \vee B \to C) \leftrightarrow (A \to C) \wedge (B \to C),$$

$$(\forall_x A \to B) \leftarrow \exists_x (A \to B) \quad \textit{if } x \notin \mathrm{FV}(B),$$
$$(A \to \forall_x B) \leftrightarrow \forall_x (A \to B) \quad \textit{if } x \notin \mathrm{FV}(A),$$

$$(\exists_x A \to B) \leftrightarrow \forall_x (A \to B) \quad \textit{if } x \notin \mathrm{FV}(B),$$
$$(A \to \exists_x B) \leftarrow \exists_x (A \to B) \quad \textit{if } x \notin \mathrm{FV}(A).$$

The proof is left as an exercise. One can also show easily that the weak variants of disjunction and the existential quantifier are indeed weaker than the proper ones:

$$A \vee B \to \mathrm{A} \mathbin{\tilde{\vee}} B, \qquad \exists_x A \to \tilde{\exists}_x A.$$

1.2.4. Intuitionistic and classical derivability. In the definition of derivability in 1.2.3 falsity $\bot$ plays no role. We may change this and require *ex-falso-quodlibet* axioms, of the form

$$\forall_{\vec{x}}(\bot \to R\vec{x}\,)$$

with R a relation symbol distinct from $\bot$. Let Efq denote the set of all such axioms. A formula A is called *intuitionistically derivable*, written $\vdash_i A$, if $\mathrm{Efq} \vdash A$. We write $\Gamma \vdash_i B$ for $\Gamma \cup \mathrm{Efq} \vdash B$.

We may even go further and require *stablity* axioms, of the form

$$\forall_{\vec{x}}(\neg\neg R\vec{x} \to R\vec{x}\,)$$

with R again a relation symbol distinct from $\bot$. Let Stab denote the set of all these axioms. A formula A is called *classically derivable*, written $\vdash_c A$, if $\mathrm{Stab} \vdash A$. We write $\Gamma \vdash_c B$ for $\Gamma \cup \mathrm{Stab} \vdash B$.

It is easy to see that intuitionistically (i.e., from Efq) we can derive $\bot \to A$ for an *arbitrary* formula A, using the introduction rules for the connectives. A similar generalization of the stability axioms is only possible for formulas in the language not involving $\vee, \exists$. However, it is still possible to use the substitutes $\tilde{\vee}$ and $\tilde{\exists}$.

THEOREM (Stability, or principle of indirect proof). *For every formula A in the language not involving $\vee, \exists$*

$$\vdash_c \neg\neg A \to A.$$

PROOF. Induction on A. For simplicity, in the derivation to be constructed we leave out applications of $\to^+$ at the end. *Case* $R\vec{t}$ with R distinct from $\bot$. Use Stab. *Case* $\bot$. The desired derivation is

$$\dfrac{v\colon (\bot \to \bot) \to \bot \qquad \dfrac{u\colon \bot}{\bot \to \bot}\,{\to^+} u}{\bot}$$

Case $A \to B$. Use $\vdash (\neg\neg B \to B) \to \neg\neg(A \to B) \to A \to B$; a derivation is

$$\dfrac{u\colon \neg\neg B \to B \qquad \dfrac{\dfrac{v\colon \neg\neg(A \to B) \qquad \dfrac{\dfrac{u_1\colon \neg B \qquad \dfrac{u_2\colon A \to B \qquad w\colon A}{B}}{\bot}}{\neg(A \to B)}\,{\to^+} u_2}{\bot}}{\neg\neg B}\,{\to^+} u_1}{B}$$

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

$$\dfrac{u\colon \neg\neg A \to A \qquad \dfrac{\dfrac{v\colon \neg\neg\forall_x A \qquad \dfrac{\dfrac{u_1\colon \neg A \qquad \dfrac{u_2\colon \forall_x A \qquad x}{A}}{\bot}}{\neg\forall_x A}\,{\to^+} u_2}{\bot}}{\neg\neg A}\,{\to^+} u_1}{A}$$

The case $A \wedge B$ is left as an exercise. □

REMARK. The argument given proves a slightly more general proposition, since in the implication case the IH was used for the conclusion only.

Using stability we can prove some well-known facts about the interaction of the weak existential quantifier with implication.

LEMMA.

$\vdash_c (\forall_x A \to B) \leftrightarrow \tilde{\exists}_x (A \to B)$ *if* $x \notin \mathrm{FV}(B)$ *and* A, B *without* $\vee, \exists$,

$\vdash_c (\tilde{\exists}_x A \to B) \leftrightarrow \forall_x (A \to B)$ *if* $x \notin \mathrm{FV}(B)$ *and* B *without* $\vee, \exists$,

$$\vdash_c (A \to \tilde{\exists}_x B) \leftrightarrow \tilde{\exists}_x (A \to B) \quad \textit{if } x \notin \mathrm{FV}(A) \textit{ and } A \textit{ without } \vee, \exists.$$

The proof is left as an exercise. Moreover, using stability we can show that the weak variants of disjunction and the existential quantifier satisfy the same axioms as the proper connectives, if one restricts the conclusion of the elimination axioms to formulas without $\vee, \exists$:

LEMMA.

$$\vdash A \to A \tilde{\vee} B,$$
$$\vdash B \to A \tilde{\vee} B,$$
$$\vdash_c A \tilde{\vee} B \to (A \to C) \to (B \to C) \to C \quad (C \textit{ without } \vee, \exists),$$

$$\vdash A \to \tilde{\exists}_x A,$$
$$\vdash_c \tilde{\exists}_x A \to \forall_x (A \to B) \to B \quad (x \notin \mathrm{FV}(B) \textit{ and } B \textit{ without } \vee, \exists).$$

Again the proof is left as an exercise.

1.3. Normalization

A derivation in normal form does not make "detours", or more precisely, it cannot occur that an elimination rule immediately follows an introduction rule. We will use "conversions" to remove such "local maxima" of complexity. However, there is a difficulty when we consider an elimination rule for $\vee$, $\wedge$ or $\exists$. An introduced formula may be used as a minor premise of an application of $\vee^-$, $\wedge^-$ 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; *permutative conversions* are designed for exactly this situation. In a permutative conversion we permute an E-rule upwards over the minor premises of $\vee^-$, $\wedge^-$ or $\exists^-$. We show that every derivation can be brought into a normal form. The shape of derivations in normal form will be analyzed. In particular, we will prove the subformula property, which says that every formula in a normal derivation is a subformula of the goal formula or else of an assumption.

It will be convenient to write derivations as terms, where the derived formula is seen as the "type" of the term. This representation is known under the name *Curry-Howard correspondence*. We give an inductive definition of derivation terms for the $\to, \forall$-rules in table 1 where for clarity we have written the corresponding derivations to the left. In table 2 this is extended also cover the rules for $\vee$, $\wedge$ and $\exists$.

1.3.1. Conversions. A conversion eliminates a detour in a derivation, i.e., an elimination immediately following an introduction. We now spell

derivation	term
$u\colon A$	u^A
$\dfrac{\begin{array}{c}[u\colon A]\\ \mid M\\ B\end{array}}{A\to B}\to^+u$	$(\lambda_{u^A}M^B)^{A\to B}$
$\dfrac{\begin{array}{c}\mid M\\ A\to B\end{array}\qquad\begin{array}{c}\mid N\\ A\end{array}}{B}\to^-$	$(M^{A\to B}N^A)^B$
$\dfrac{\begin{array}{c}\mid M\\ A\end{array}}{\forall_x A}\forall^+$ (with var.cond.)	$(\lambda_x M^A)^{\forall_x A}$ (with var.cond.)
$\dfrac{\begin{array}{c}\mid M\\ \forall_x A(x)\end{array}\qquad r}{A(r)}\forall^-$	$(M^{\forall_x A(x)}r)^{A(r)}$

TABLE 1. Derivation terms for $\to$ and $\forall$

out in detail which conversions we shall allow. This is done for derivations written in tree notation and also as derivation terms.

$\to$*-conversion.*

$$\dfrac{\dfrac{\begin{array}{c}[u\colon A]\\ \mid M\\ B\end{array}}{A\to B}\to^+u \qquad \begin{array}{c}\mid N\\ A\end{array}}{B}\to^- \quad\mapsto\quad \begin{array}{l}\mid N\\ A\\ \mid M\\ B\end{array}$$

or written as derivation terms $(\lambda_u M(u^A)^B)^{A\to B}N^A \mapsto M(N^A)^B$.

derivation	term
$\dfrac{\begin{matrix}\mid M\\ A\end{matrix}}{A\vee B}\vee_0^+ \qquad \dfrac{\begin{matrix}\mid M\\ B\end{matrix}}{A\vee B}\vee_1^+$	$(\vee_{0,B}^+ M^A)^{A\vee B} \quad (\vee_{1,A}^+ M^B)^{A\vee B}$
$\dfrac{\begin{matrix}\\ \mid M\\ A\vee B\end{matrix}\quad\begin{matrix}[u\colon A]\\ \mid N\\ C\end{matrix}\quad\begin{matrix}[v\colon B]\\ \mid K\\ C\end{matrix}}{C}\vee^- u,v$	$(M^{A\vee B}(u^A.N^C, v^B.K^C))^C$
$\dfrac{\begin{matrix}\mid M\\ A\end{matrix}\quad\begin{matrix}\mid N\\ B\end{matrix}}{A\wedge B}\wedge^+$	$\langle M^A, N^B\rangle^{A\wedge B}$
$\dfrac{\begin{matrix}\\ \mid M\\ A\wedge B\end{matrix}\quad\begin{matrix}[u\colon A]\quad[v\colon B]\\ \mid N\\ C\end{matrix}}{C}\wedge^- u,v$	$(M^{A\wedge B}(u^A, v^B.N^C))^C$
$\dfrac{\begin{matrix}\\ r\end{matrix}\quad\begin{matrix}\mid M\\ A(r)\end{matrix}}{\exists_x A(x)}\exists^+$	$(\exists_{x,A}^+ r M^{A(r)})^{\exists_x A(x)}$
$\dfrac{\begin{matrix}\\ \mid M\\ \exists_x A\end{matrix}\quad\begin{matrix}[u\colon A]\\ \mid N\\ B\end{matrix}}{B}\exists^- x,u$ (var.cond.)	$(M^{\exists_x A}(u^A.N^B))^B$ (var.cond.)

TABLE 2. Derivation terms for $\vee$, $\wedge$ and $\exists$

$\forall$-*conversion.*

$$\dfrac{\dfrac{\begin{array}{c}\mid M\\ A(x)\end{array}}{\forall_x A(x)}\,\forall^+ \qquad r}{A(r)}\,\forall^- \quad\mapsto\quad \begin{array}{c}\mid M'\\ A(r)\end{array}$$

or written as derivation terms $(\lambda_x M(x)^{A(x)})^{\forall_x A(x)} r \mapsto M(r)$.

$\vee$-*conversion.*

$$\dfrac{\dfrac{\begin{array}{c}\mid M\\ A\end{array}}{A\vee B}\,\vee_0^+ \qquad \begin{array}{c}[u\colon A]\\ \mid N\\ C\end{array} \qquad \begin{array}{c}[v\colon B]\\ \mid K\\ C\end{array}}{C}\,\vee^- u,v \quad\mapsto\quad \begin{array}{c}\mid M\\ A\\ \mid N\\ C\end{array}$$

or as derivation terms $(\vee_{0,B}^+ M^A)^{A\vee B}(u^A.N(u)^C, v^B.K(v)^C) \mapsto N(M^A)^C$, and

$$\dfrac{\dfrac{\begin{array}{c}\mid M\\ B\end{array}}{A\vee B}\,\vee_1^+ \qquad \begin{array}{c}[u\colon A]\\ \mid N\\ C\end{array} \qquad \begin{array}{c}[v\colon B]\\ \mid K\\ C\end{array}}{C}\,\vee^- u,v \quad\mapsto\quad \begin{array}{c}\mid M\\ B\\ \mid K\\ C\end{array}$$

$\wedge$-*conversion.*

$$\dfrac{\dfrac{\begin{array}{c}\mid M\\ A\end{array} \qquad \begin{array}{c}\mid N\\ B\end{array}}{A\wedge B}\,\wedge^+ \qquad \begin{array}{c}[u\colon A]\quad [v\colon B]\\ \mid K\\ C\end{array}}{C}\,\wedge^- u,v \quad\mapsto\quad \begin{array}{c}\begin{array}{c}\mid M\\ A\end{array} \qquad \begin{array}{c}\mid N\\ B\end{array}\\ \mid K\\ C\end{array}$$

or $\langle M^A, N^B\rangle^{A\wedge B}(u^A, v^B.K(u,v)^C) \mapsto K(M^A, N^B)^C$.

$\exists$-*conversion.*

$$\dfrac{\dfrac{r \qquad \begin{array}{c}\mid M\\ A(r)\end{array}}{\exists_x A(x)}\,\exists^+ \qquad \begin{array}{c}[u\colon A(x)]\\ \mid N\\ B\end{array}}{B}\,\exists^- x,u \quad\mapsto\quad \begin{array}{c}\mid M\\ A(r)\\ \mid N'\\ B\end{array}$$

or $(\exists_{x,A}^+ r M^{A(r)})^{\exists_x A(x)}(u^{A(x)}.N(x,u)^B) \mapsto N(r, M^{A(r)})^B$.

We now consider the permutative conversions.

$\vee$-*permutative conversion.*

$$\dfrac{\dfrac{\begin{matrix}\mid M\\ A\vee B\end{matrix}\qquad\begin{matrix}\mid N\\ C\end{matrix}\qquad\begin{matrix}\mid K\\ C\end{matrix}}{C}\qquad\begin{matrix}\mid L\\ C'\end{matrix}}{D}\,\text{E-rule}\qquad\mapsto$$

$$\dfrac{\begin{matrix}\mid M\\ A\vee B\end{matrix}\qquad\dfrac{\begin{matrix}\mid N\\ C\end{matrix}\qquad\begin{matrix}\mid L\\ C'\end{matrix}}{D}\,\text{E-rule}\qquad\dfrac{\begin{matrix}\mid K\\ C\end{matrix}\qquad\begin{matrix}\mid L\\ C'\end{matrix}}{D}\,\text{E-rule}}{D}$$

or with for instance $\to^-$ as E-rule $(M^{A\vee B}(u^A.N^{C\to D}, v^B.K^{C\to D}))^{C\to D}L^C \mapsto (M^{A\vee B}(u^A.(N^{C\to D}L^C)^D, v^B.(K^{C\to D}L^C)^D))^D$.

$\wedge$-*permutative conversion.*

$$\dfrac{\dfrac{\begin{matrix}\mid M\\ A\wedge B\end{matrix}\qquad\begin{matrix}\mid N\\ C\end{matrix}}{C}\qquad\begin{matrix}\mid K\\ C'\end{matrix}}{D}\,\text{E-rule}\qquad\mapsto$$

$$\dfrac{\begin{matrix}\mid M\\ A\wedge B\end{matrix}\qquad\dfrac{\begin{matrix}\mid N\\ C\end{matrix}\qquad\begin{matrix}\mid K\\ C'\end{matrix}}{D}\,\text{E-rule}}{D}$$

or $(M^{A\wedge B}(u^A, v^B.N^{C\to D}))^{C\to D}K^C \mapsto (M^{A\wedge B}(u^A, v^B.(N^{C\to D}K^C)^D))^D$.

$\exists$-*permutative conversion.*

$$\dfrac{\dfrac{\begin{matrix}\mid M\\ \exists_x A\end{matrix}\qquad\begin{matrix}\mid N\\ B\end{matrix}}{B}\qquad\begin{matrix}\mid K\\ C\end{matrix}}{D}\,\text{E-rule}\qquad\mapsto$$

$$\dfrac{\begin{matrix}\mid M\\ \exists_x A\end{matrix}\qquad\dfrac{\begin{matrix}\mid N\\ B\end{matrix}\qquad\begin{matrix}\mid K\\ C\end{matrix}}{D}\,\text{E-rule}}{D}$$

or $(M^{\exists_x A}(u^A.N^{C\to D}))^{C\to D}K^C \mapsto (M^{\exists_x A}(u^A.(N^{C\to D}K^C)^D))^D$.

We will use further somewhat trivial conversions, which remove unnecessary applications of the elimination rules for $\vee$, $\wedge$ and $\exists$. They are called *simplification conversions*. For $\vee$ we have

$$\dfrac{\begin{array}{c}\mid M\\ A\vee B\end{array}\qquad \begin{array}{c}[u\colon A]\\ \mid N\\ C\end{array}\qquad \begin{array}{c}[v\colon B]\\ \mid K\\ C\end{array}}{C}\,\vee^- u,v \qquad\mapsto\qquad \begin{array}{c}\mid N\\ C\end{array}$$

if $u\colon A$ is not free in N, or $(M^{A\vee B}(u^A.N^C, v^B.K^C))^C \mapsto N^C$; similar for the second component. For $\wedge$ there is the conversion

$$\dfrac{\begin{array}{c}\mid M\\ A\wedge B\end{array}\qquad \begin{array}{c}[u\colon A]\quad [v\colon B]\\ \mid N\\ C\end{array}}{C}\,\wedge^- u,v \qquad\mapsto\qquad \begin{array}{c}\mid N\\ C\end{array}$$

if neither $u\colon A$ nor $v\colon B$ is free in N, or $(M^{A\wedge B}(u^A, v^B.N^C))^C \mapsto N^C$. For $\exists$ the simplification conversion is

$$\dfrac{\begin{array}{c}\mid M\\ \exists_x A\end{array}\qquad \begin{array}{c}[u\colon A]\\ \mid N\\ B\end{array}}{B}\,\exists^- x,u \qquad\mapsto\qquad \begin{array}{c}\mid N\\ B\end{array}$$

if again $u\colon A$ is not free in N, or $(M^{\exists_x A}(u^A.N^B))^B \mapsto N^B$.

1.3.2. Strong normalization. We now show that no matter in which order we apply the conversion rules, they will always terminate and produce a derivation in "normal form", where no further conversions can be applied.

We shall write derivation terms without formula superscripts, and usually leave implicit the extra (formula) parts of derivation constants. For instance, we write $\exists^+$ instead of $\exists^+_{x,A}$. Hence we consider derivation terms M, N, K of the forms

$$\begin{aligned} &u \mid \lambda_v M \mid \lambda_y M \mid \vee_0^+ M \mid \vee_1^+ M \mid \langle M, N\rangle \mid \exists^+ rM \mid \\ &\quad MN \mid Mr \mid M(v_0.N_0, v_1.N_1) \mid M(v, w.N) \mid M(v.N); \end{aligned}$$

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

To simplify the technicalities, we restrict our treatment to the rules for $\to$ and $\exists$. It can easily be extended to the full set of rules. Hence we consider

$$u \mid \lambda_v M \mid \exists^+ rM \mid MN \mid M(v.N).$$

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:

$$u\vec{M} \mid u\vec{M}E \mid \lambda_v M \mid \exists^+ rM \mid$$
$$(\lambda_v M)N\vec{R} \mid \exists^+ rM(v.N)\vec{R} \mid u\vec{M}ER\vec{S}.$$

Here the final three forms are not normal: $(\lambda_v M)N\vec{R}$ and $\exists^+ rM(v.N)\vec{R}$ both are *β-redexes*, and $u\vec{M}ER\vec{S}$ is a *permutative redex*. The conversion rules are

$$\begin{array}{lll} (\lambda_v M(v))N & \mapsto_\beta M(N) & \beta_\to\text{-conversion}, \\ \exists^+_{x,A} rM(v.N(x,v)) & \mapsto_\beta N(r,M) & \beta_\exists\text{-conversion}, \\ M(v.N)R & \mapsto_\pi M(v.NR) & \text{permutative conversion.} \end{array}$$

In addition we also allow

$$M(v.N) \mapsto_\sigma N \quad \text{if } v\colon A \text{ is not free in } N\text{; simplification conversion.}$$

$M(v.N)$ is a *simplification redex*. The *closure* of these conversions is defined by

(a) If $M \mapsto_\xi M'$ for $\xi = \beta, \pi, \sigma$, then $M \to M'$.
(b) If $M \to M'$, then $MR \to M'R$, $NM \to NM'$, $N(v.M) \to N(v.M')$, $\lambda_v M \to \lambda_v M'$, $\exists^+ rM \to \exists^+ rM'$ (*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 conversum M'' of M', i.e., by a single conversion. The relation $\to^+$ ("*properly reduces to*") is the transitive closure of $\to$, and $\to^*$ ("*reduces to*") is the reflexive and transitive closure of $\to$. 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 $M \to^* N$. A *reduction sequence* is a (finite or infinite) sequence $M_0 \to M_1 \to M_2 \to \dots$ such that $M_i \to M_{i+1}$, for all i.

We inductively define a set SN. In doing so we take care that for a given M there is exactly one rule applicable to generate $M \in \mathrm{SN}$. This will be crucial to make the later proofs work.

$$\frac{\vec{M} \in \mathrm{SN}}{u\vec{M} \in \mathrm{SN}}\,(\mathrm{Var}_0) \qquad \frac{M \in \mathrm{SN}}{\lambda_v M \in \mathrm{SN}}\,(\lambda) \qquad \frac{M \in \mathrm{SN}}{\exists^+ rM \in \mathrm{SN}}\,(\exists)$$

$$\frac{\vec{M}, N \in \mathrm{SN}}{u\vec{M}(v.N) \in \mathrm{SN}}\,(\mathrm{Var}) \qquad \frac{u\vec{M}(v.NR)\vec{S} \in \mathrm{SN}}{u\vec{M}(v.N)R\vec{S} \in \mathrm{SN}}\,(\mathrm{Var}_\pi)$$

$$\frac{M(N)\vec{R} \in \mathrm{SN} \qquad N \in \mathrm{SN}}{(\lambda_v M(v))N\vec{R} \in \mathrm{SN}}\,(\beta_\to)$$

$$\frac{N(r,M)\vec{R} \in \mathrm{SN} \qquad M \in \mathrm{SN}}{\exists^+_{x,A} r M(v.N(x,v))\vec{R} \in \mathrm{SN}}\,(\beta_\exists)$$

where in (Var_π) we require that x (from $\exists_x A$) and v are not free in R.

It is easy to see that SN is closed under substitution for object variables: if $M(x) \in \mathrm{SN}$, then $M(r) \in \mathrm{SN}$. The proof is by induction on $M \in \mathrm{SN}$. In each case the same rule can be applied again.

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

Lemma. *Every term in* SN *is strongly normalizing.*

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

Case (Var_π). We prove, as an auxiliary lemma, that

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

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

$$u\vec{M}(v.N(v'.N'))T\vec{T}$$

leads in two permutative steps to M, hence by assumption $M\downarrow$.

Case ($\beta_\to$). We show that $M(N)\vec{R}\downarrow$ and $N\downarrow$ imply $(\lambda_v M(v))N\vec{R}\downarrow$. This is done by a induction on $N\downarrow$, with a side induction on $M(N)\vec{R}\downarrow$. We need to consider all possible reducts of $(\lambda_v M(v))N\vec{R}$. In case of an outer β-reduction use the assumption. If N is reduced, use the IH. Reductions in M and in $\vec{R}$ as well as permutative reductions within $\vec{R}$ are taken care of by the side IH.

Case ($\beta_\exists$). We show that

$$N(r,M)\vec{R}\downarrow \text{ and } M\downarrow \text{ together imply } \exists^+ r M(v.N(x,v))\vec{R}\downarrow.$$

This is done by a threefold induction: first on $M\downarrow$, second on $N(r,M)\vec{R}\downarrow$ and third on the length of $\vec{R}$. We need to consider all possible reducts of $\exists^+ r M(v.N(x,v))\vec{R}$. In case of an outer β-reduction use the assumption. If M is reduced, use the first IH. Reductions in $N(x,v)$ and in $\vec{R}$ as well as permutative reductions within $\vec{R}$ are taken care of by the second IH. The only remaining case is when $\vec{R} = S\vec{S}$ and $(v.N(x,v))$ is permuted with S, to yield $\exists^+ r M(v.N(x,v)S)\vec{S}$. Apply the third IH. $\square$

For later use we prove a slightly generalized form of the rule (Var_π):

Proposition. *If* $M(v.NR)\vec{S} \in \mathrm{SN}$*, then* $M(v.N)R\vec{S} \in \mathrm{SN}$.

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

Case $u\vec{T}(v.NR)\vec{S} \in \mathrm{SN}$. If $\vec{T} = \vec{M}$, use (Var_π). Otherwise we have $u\vec{M}(v'.N')\vec{R}(v.NR)\vec{S} \in \mathrm{SN}$. This must be generated by repeated applications of (Var_π) from $u\vec{M}(v'.N'\vec{R}(v.NR)\vec{S}) \in \mathrm{SN}$, and finally by (Var) from $\vec{M} \in \mathrm{SN}$ and $N'\vec{R}(v.NR)\vec{S} \in \mathrm{SN}$. The IH for the latter yields $N'\vec{R}(v.N)R\vec{S} \in \mathrm{SN}$, hence $u\vec{M}(v'.N'\vec{R}(v.N)R\vec{S}) \in \mathrm{SN}$ by (Var) and finally $u\vec{M}(v'.N')\vec{R}(v.N)R\vec{S} \in \mathrm{SN}$ by (Var_π).

Case $\exists^+ rM\vec{T}(v.N(x,v)R)\vec{S} \in \mathrm{SN}$. Similar, with ($\beta_\exists$) instead of ($\mathrm{Var}_\pi$). In detail: If $\vec{T}$ is empty, by ($\beta_\exists$) this came from $N(r,M)R\vec{S} \in \mathrm{SN}$ and $M \in \mathrm{SN}$, hence $\exists^+ rM(v.N(x,v))R\vec{S} \in \mathrm{SN}$ again by ($\beta_\exists$). Otherwise we have $\exists^+ rM(v'.N'(x',v'))\vec{T}(v.NR)\vec{S} \in \mathrm{SN}$. This must be generated by ($\beta_\exists$) from $N'(r,M)\vec{T}(v.NR)\vec{S} \in \mathrm{SN}$. The IH yields $N'(r,M)\vec{T}(v.N)R\vec{S} \in \mathrm{SN}$, hence $\exists^+ rM(v'.N'(x,v'))\vec{T}(v.N)R\vec{S} \in \mathrm{SN}$ by ($\beta_\exists$).

Case $(\lambda_v M(v))N'\vec{R}(w.NR)\vec{S} \in \mathrm{SN}$. By ($\beta_\to$) this came from $N' \in \mathrm{SN}$ and $M(N')\vec{R}(w.NR)\vec{S} \in \mathrm{SN}$. The IH yields $M(N')\vec{R}(w.N)R\vec{S} \in \mathrm{SN}$, hence $(\lambda_v M(v))N'\vec{R}(w.N)R\vec{S} \in \mathrm{SN}$ by ($\beta_\to$). □

In what follows we shall show that *every* term is in SN and hence is strongly normalizing. Given the definition of SN we only have to show that SN is closed under $\to^-$ and $\exists^-$. 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* $M \in \mathrm{SN}$*, if* M *proves* $A = A_0 \to A_1$ *and* $N \in \mathrm{SN}$*, then* $MN \in \mathrm{SN}$*,*
(b) *for all* $M \in \mathrm{SN}$*, if* M *proves* $A = \exists_x B$ *and* $N \in \mathrm{SN}$*, then* $M(v.N) \in \mathrm{SN}$*,*
(c) *for all* $M(v) \in \mathrm{SN}$*, if* $N^A \in \mathrm{SN}$*, then* $M(N) \in \mathrm{SN}$.

Proof. Induction on $|A|$. 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 side induction on $M \in \mathrm{SN}$. Let $M \in \mathrm{SN}$ and assume that M proves $A = A_0 \to A_1$ and $N \in \mathrm{SN}$. We distinguish cases according to how $M \in \mathrm{SN}$ was generated. For (Var_0), (Var_π), ($\beta_\to$) and ($\beta_\exists$) use the same rule again.

Case $u\vec{M}(v.N') \in \mathrm{SN}$ by (Var) from $\vec{M}, N' \in \mathrm{SN}$. Then $N'N \in \mathrm{SN}$ by SIH for N', hence $u\vec{M}(v.N'N) \in \mathrm{SN}$ by (Var), hence $u\vec{M}(v.N')N \in \mathrm{SN}$ by (Var_π).

Case $(\lambda_v M(v))^{A_0 \to A_1} \in \mathrm{SN}$ by (λ) from $M \in \mathrm{SN}$. Use $(\beta_\to)$; for this we need to know $M(N) \in \mathrm{SN}$. But this follows from IH(c) for M, since N derives A_0.

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

Case $u\vec{M} \in \mathrm{SN}$ by (Var_0) from $\vec{M} \in \mathrm{SN}$. Use (Var).

Case $(\exists^+ rM)^{\exists_x A} \in \mathrm{SN}$ by $(\exists)$ from $M \in \mathrm{SN}$. We must show that $\exists^+ rM(v.N(x,v)) \in \mathrm{SN}$. Use $(\beta_\exists)$; for this we need to know $N(r,M) \in \mathrm{SN}$. But this follows from IH(c) for $N(r,v)$ (which is in SN by the remark above), since M derives $A(r)$.

Case $u\vec{M}(v'.N') \in \mathrm{SN}$ by (Var) from $\vec{M}, N' \in \mathrm{SN}$. Then $N'(v.N) \in \mathrm{SN}$ by SIH for N', hence $u\vec{M}(v.N'(v.N)) \in \mathrm{SN}$ by (Var) and therefore $u\vec{M}(v.N')(v.N) \in \mathrm{SN}$ by (Var_π).

(c). By side induction on $M(v) \in \mathrm{SN}$. Let $N^A \in \mathrm{SN}$; the goal is $M(N) \in \mathrm{SN}$. We distinguish cases according to how $M(v) \in \mathrm{SN}$ was generated. For (λ), $(\exists)$, $(\beta_\to)$ and $(\beta_\exists)$ use the same rule again.

Case $u\vec{M}(v) \in \mathrm{SN}$ by (Var_0) from $\vec{M}(v) \in \mathrm{SN}$. Then $\vec{M}(N) \in \mathrm{SN}$ by SIH(c). If $u \neq v$, use (Var_0) again. If $u = v$, we must show $N\vec{M}(N) \in \mathrm{SN}$. Note that N proves A; hence the claim follows from $\vec{M}(N) \in \mathrm{SN}$ by (a).

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

Case $u\vec{M}(v)(v'.N'(v))R(v)\vec{S}(v) \in \mathrm{SN}$ has been obtained by (Var_π) from $u\vec{M}(v)(v'.N'(v)R(v))\vec{S}(v) \in \mathrm{SN}$. If $u \neq v$, use (Var_π) again. If $u = v$, from the IH we obtain $N\vec{M}(N)(v'.N'(N)R(N))\vec{S}(N) \in \mathrm{SN}$. Now use the proposition above. □

COROLLARY. *Every term is strongly normalizing.*

PROOF. Induction on the (first) inductive definition of terms M. In cases u, $\lambda_v M$ and $\exists^+ rM$ the claim follows from the definition of SN, and in cases MN and $M(v.N)$ from parts (a), (b) of the previous theorem. □

1.3.3. The structure of normal derivations. To analyze normal derivations, it will be useful to introduce the notions of a *segment* and of a *track* in a proof tree, which make sense for non-normal derivations as well.

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 $1 \le i < n$, 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 $\exists^-$.

(Notice that a formula occurrence 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 use σ, σ' for segments. σ is called a *subformula* of σ' if the formula A in σ is a subformula of B in σ'.

The 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 $A_0, \dots, A_n$ 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 $\to^-$, 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 $\to^-$, *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.

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

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

$$\dfrac{\begin{array}{c} \mid M \\ \exists_x A \end{array} \qquad \begin{array}{c} [u\colon A] \\ \mid N \\ B \end{array}}{B}\ \exists^- x, u$$

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

Definition. A *track of order* 0, or *main track*, in a derivation is a track ending either in the conclusion of the whole derivation or in the major premise of an application of a $\vee^-$, $\wedge^-$ 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 π (i.e., a linearly ordered subtree) in the proof tree such that π passes only through premises of I-rules and *major premises* of E-rules, and π begins at a top node and ends in the conclusion.

Since by simplification conversions we have removed every application of an $\vee^-$, $\wedge^-$ or $\exists^-$-rule that discharges no assumption variables, each track of order 0 in a normal derivation is a track ending in the conclusion of the whole derivation. Note also that 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 $\wedge^+$-application. Now let us consider normal derivations. Recall the notion of a strictly positive part of a formula, defined in 1.1.4.

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

(a) *for each σ_j in the E-part one has $j < i$, σ_j is a major premise of an E-rule, and σ_{j+1} is a strictly positive part of σ_j, and therefore each σ_j is a s.p.p. of σ_0;*
(b) *for each σ_j which is the minimum segment or is in the I-part one has $i \le j$, and if $j \ne n$, then σ_j is a premise of an I-rule and a s.p.p. of σ_{j+1}, so each σ_j is a s.p.p. of σ_n.*

Theorem (Subformula property). *Let M be a normal derivation. Then each formula occurring in the derivation is a subformula of either the end formula or else an assumption formula.*

Proof. As noted 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 Γ does not contain a disjunction as s.p.p. (strictly positive part), then, if $\Gamma \vdash A \vee B$, it follows that $\Gamma \vdash A$ or $\Gamma \vdash B$.*

Proof. Consider a normal derivation M of $A \vee B$ from assumptions Γ not containing a disjunction as s.p.p. The conclusion $A \vee B$ is the final formula of a (main) track, whose top formula A_0 in M must be an assumption in Γ. Since Γ does not contain a disjunction as s.p.p., the segment σ with the conclusion $A \vee B$ is in the I-part. Skip the final $\vee_i^+$-rule and replace the formulas in σ 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* $\Gamma \vdash \exists_x A(x)$.

(a) *If* Γ *does not contain an existential s.p.p., then there are terms* $r_1, \dots, r_n$ *such that* $\Gamma \vdash A(r_1) \vee \dots \vee A(r_n)$.
(b) *If* Γ *neither contains a disjunctive s.p.p., nor an existential s.p.p., then there is a term* r *such that* $\Gamma \vdash A(r)$.

Proof. Consider a normal derivation M of $\exists_x A(x)$ from assumptions Γ 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 $\vee^-$ and leave the others to the reader.

$$\dfrac{\begin{matrix} \\ \mid M \\ B \vee C \end{matrix} \qquad \begin{matrix} [u\colon B] \\ \mid N_0 \\ \exists_x A(x) \end{matrix} \qquad \begin{matrix} [v\colon C] \\ \mid N_1 \\ \exists_x A(x) \end{matrix}}{\exists_x A(x)} \vee^- u, v$$

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

$$\dfrac{\begin{matrix} \\ \\ \mid M \\ B \vee C \end{matrix} \qquad \begin{matrix} [u\colon B] \\ \mid \\ \dfrac{\bigvee_{i=1}^{n} A(r_i)}{\bigvee_{i=1}^{n+m} A(r_i)} \vee^+ \end{matrix} \qquad \begin{matrix} [v\colon C] \\ \mid \\ \dfrac{\bigvee_{i=n+1}^{n+m} A(r_i)}{\bigvee_{i=1}^{n+m} A(r_i)} \vee^+ \end{matrix}}{\bigvee_{i=1}^{n+m} A(r_i)} \vee^- u, v$$

(b). Similarly; by assumption the last rule can be neither $\vee^-$ nor $\exists^-$. □

1.4. Soundness and Completeness for Tree Models

It is an obvious question to ask whether the logical rules we have been considering suffice, i.e., whether we have forgotten some necessary rules. To answer this question we first have to fix the *meaning* of a formula, i.e., provide a semantics. This will be done by means of tree models. Using this concept of a model we will prove soundness and completeness.

1.4.1. Tree models. Consider a finitely branching tree of "possible worlds". The worlds are represented as nodes in this tree. They may be thought of as possible states such that all nodes "above" a node k are the ways in which k may develop in the future. The worlds are increasing, that is, if an atomic formula $R\vec{s}$ is true in a world k, then $R\vec{s}$ is true in all future worlds k'.

More formally, each tree model is based on a finitely branching tree T. A node k over a set S is a finite sequence $k = \langle a_0, a_1, \dots, a_{n-1} \rangle$ of elements of S; $\mathrm{lh}(k)$ is the length of k. We write $k \preceq k'$ if k is an initial segment of k'. A tree on S is a set of nodes closed under initial segments. A tree T is finitely branching if every node in T has finitely many immediate successors. A tree T is *infinite* if for every $n \in \mathbb{N}$ there is a node $k \in T$ such that $\mathrm{lh}(k) = n$. A *branch* of T is a linearly ordered subtree of T. A *leaf* is a node without successors in T.

For the proof of the completeness theorem, a tree model based on a complete binary tree (i.e., the complete tree over $\{0, 1\}$) will suffice. The nodes will be all the finite sequences of 0's and 1's, and the ordering is as above. The root is the empty sequence and $k0$ is the sequence k with the element 0 added at the end; similarly for $k1$.

For the rest of this section, fix a countable formal language $\mathcal{L}$; we do not mention the dependence on $\mathcal{L}$ in the notation.

Definition. Let T be a finitely branching tree. A *tree model* on T is a triple $\mathcal{T} = (D, I_0, I_1)$ such that

(a) D is a nonempty set;
(b) for every n-ary function symbol f (in the underlying language $\mathcal{L}$), I_0 assigns to f a map $I_0(f)\colon D^n \to D$;
(c) for every n-ary relation symbol R and every node $k \in T$, $I_1(R, k) \subseteq D^n$ is assigned in such a way that monotonicity is preserved:

$$k \preceq k' \to I_1(R, k) \subseteq I_1(R, k').$$

If $n = 0$, then $I_1(R, k)$ is either true or false, and by monotonicity it follows that if $k \preceq k'$ and $I_1(R, k)$ then $I_1(R, k')$. There is no special requirement set on $I_1(\bot, k)$. (Recall that falsum $\bot$ is viewed as an ordinary propositional symbol.) We write $R^{\mathcal{T}}(\vec{a}, k)$ for $\vec{a} \in I_1(R, k)$, and $|\mathcal{T}|$ for D.

It is obvious from the definition that any tree T can be extended to a complete tree $\bar{T}$ without leaves, in which for every leaf $k \in T$ all sequences $k0$, $k00$, $k000, \dots$ are added to T. For every node $k0 \dots 0$, we then add $I_1(R, k0 \dots 0) := I_1(R, k)$.

An *assignment* (or variable assignment) in D is a map η assigning to every variable $x \in \mathrm{dom}(\eta)$ a value $\eta(x) \in D$. Finite assignments will be written as $[x_1 := a_1, \dots, x_n := a_n]$ (or else as $[a_1/x_1, \dots, a_n/x_n]$), with distinct $x_1, \dots, x_n$. If η is an assignment in D and $a \in D$, let η_x^a be the assignment in D mapping x to a and coinciding with η elsewhere:

$$\eta_x^a(y) := \begin{cases} \eta(y), & \text{if } y \neq x \\ a, & \text{if } y = x. \end{cases}$$

Let a tree model $\mathcal{T} = (D, I_0, I_1)$ and an assignment η in D be given. We define a homomorphic extension of η (denoted by η as well) to terms t such that $\mathrm{vars}(t) \subseteq \mathrm{dom}(\eta)$ by

$$\begin{aligned} \eta(c) &:= I_0(c), \\ \eta(f(t_1, \dots, t_n)) &:= I_0(f)(\eta(t_1), \dots, \eta(t_n)). \end{aligned}$$

Observe that the extension of η depends on $\mathcal{T}$; we often write $t^{\mathcal{T}}[\eta]$ for $\eta(t)$.

Definition. $\mathcal{T}, k \Vdash A[\eta]$ (*$\mathcal{T}$ forces A at node k for an assignment η*) is defined inductively. We write $k \Vdash A[\eta]$ when it is clear from the context what the underlying model $\mathcal{T}$ is, and $\forall_{k' \succeq_n k} A$ for $\forall_{k' \succeq k}(\mathrm{lh}(k') = \mathrm{lh}(k) + n \to A)$.

$$\begin{aligned} k \Vdash (R\vec{s})[\eta] &:= \exists_n \forall_{k' \succeq_n k} R^{\mathcal{T}}(\vec{s}^{\mathcal{T}}[\eta], k'). \\ k \Vdash (A \vee B)[\eta] &:= \exists_n \forall_{k' \succeq_n k}(k' \Vdash A[\eta] \vee k' \Vdash B[\eta]). \\ k \Vdash (\exists_x A)[\eta] &:= \exists_n \forall_{k' \succeq_n k} \exists_{a \in |\mathcal{T}|}(k' \Vdash A[\eta_x^a]). \\ k \Vdash (A \to B)[\eta] &:= \forall_{k' \succeq k}(k' \Vdash A[\eta] \to k' \Vdash B[\eta]). \\ k \Vdash (A \wedge B)[\eta] &:= k \Vdash A[\eta] \wedge k \Vdash B[\eta]. \\ k \Vdash (\forall_x A)[\eta] &:= \forall_{a \in |\mathcal{T}|}(k \Vdash A[\eta_x^a]). \end{aligned}$$

In this definition, the logical connectives $\to, \wedge, \vee, \forall, \exists$ on the left hand side are part of the object language, whereas the same connectives on the right hand side are to be *understood* in the usual sense: they belong to the "metalanguage". It should always be clear from the context whether a formula is part of the object or the metalanguage.

Note that the clauses for atoms, disjunction and existential quantifier include a concept of a "bar", in $\bar{T}$.

1.4.2. Covering lemma. It is easily seen (using the definition and monotonicity) that from $k \Vdash A[\eta]$ and $k \preceq k'$ we can conclude $k' \Vdash A[\eta]$. The converse is also true:

Lemma (Covering).

$$\forall_{k' \succeq_n k}(k' \Vdash A[\eta]) \to k \Vdash A[\eta].$$

Proof. Induction on A. We write $k \Vdash A$ for $k \Vdash A[\eta]$.
Case $R\vec{s}$. Assume

$$\forall_{k' \succeq_n k}(k' \Vdash R\vec{s}),$$

hence by definition

$$\forall_{k'\succeq_n k}\exists_m\forall_{k''\succeq_m k'} R^{\mathcal{T}}(\vec{s}^{\mathcal{T}}[\eta], k'').$$

Since T is a finitely branching tree,

$$\exists_m\forall_{k'\succeq_m k} R^{\mathcal{T}}(\vec{s}^{\mathcal{T}}[\eta], k').$$

Hence $k \Vdash R\vec{s}$.

The cases $A \vee B$ and $\exists_x A$ are handled similarly.

Case $A \to B$. Let $k' \Vdash A \to B$ for all $k' \succeq k$ with $\mathrm{lh}(k') = \mathrm{lh}(k) + n$. We show

$$\forall_{l\succeq k}(l \Vdash A \to l \Vdash B).$$

Let $l \succeq k$ and $l \Vdash A$. We must show $l \Vdash B$. To this end we apply the IH to B and $m := \max(\mathrm{lh}(k) + n, \mathrm{lh}(l))$. So assume $l' \succeq l$ and $\mathrm{lh}(l') = m$. It is sufficient to show $l' \Vdash B$. If $\mathrm{lh}(l') = \mathrm{lh}(l)$, then $l' = l$ and we are done. If $\mathrm{lh}(l') = \mathrm{lh}(k) + n > \mathrm{lh}(l)$, then l' is an extension of l as well as of k and has length $\mathrm{lh}(k) + n$, and hence $l' \Vdash A \to B$ by assumption. Moreover, $l' \Vdash A$, since $l' \succeq l$ and $l \Vdash A$. It follows that $l' \Vdash B$.

The cases $A \wedge B$ and $\forall_x A$ are easy. □

1.4.3. Soundness.

Lemma (Coincidence). *Let $\mathcal{T}$ be a tree model, t a term, A a formula and η, ξ assignments in $|\mathcal{T}|$.*

(a) *If $\eta(x) = \xi(x)$ for all $x \in \mathrm{vars}(t)$, then $\eta(t) = \xi(t)$.*
(b) *If $\eta(x) = \xi(x)$ for all $x \in \mathrm{FV}(A)$, then $\mathcal{T}, k \Vdash A[\eta]$ if and only if $\mathcal{T}, k \Vdash A[\xi]$.*

Proof. Induction on terms and formulas. □

Lemma (Substitution). *Let $\mathcal{T}$ be a tree model, t, r terms, A a formula and η an assignment in $|\mathcal{T}|$. Then*

(a) $\eta(r(t)) = \eta_x^{\eta(t)}(r(x))$.
(b) *$\mathcal{T}, k \Vdash A(t)[\eta]$ if and only if $\mathcal{T}, k \Vdash A(x)[\eta_x^{\eta(t)}]$.*

Proof. Induction on terms and formulas. □

Theorem (Soundness). *Let $\Gamma \cup \{A\}$ be a set of formulas such that $\Gamma \vdash A$. Then, if $\mathcal{T}$ is a tree model, k a node and η an assignment in $|\mathcal{T}|$, it follows that $\mathcal{T}, k \Vdash \Gamma[\eta]$ entails $\mathcal{T}, k \Vdash A[\eta]$.*

Proof. Induction on derivations.

We begin with the axiom schemes $\vee_0^+$, $\vee_1^+$, $\vee^-$, $\wedge^+$, $\wedge^-$, $\exists^+$ and $\exists^-$. $k \Vdash C[\eta]$ is abbreviated $k \Vdash C$, when η is known from the context.

Case $\vee_0^+$: $A \to A \vee B$. We show $k \Vdash A \to A \vee B$. Assume for $k' \succeq k$ that $k' \Vdash A$. Show: $k' \Vdash A \vee B$. This follows from the definition, since $k' \Vdash A$. The case $\vee_1^+$: $B \to A \vee B$ is symmetric.

Case $\vee^-$: $A \vee B \to (A \to C) \to (B \to C) \to C$. We show that $k \Vdash A \vee B \to (A \to C) \to (B \to C) \to C$. Assume for $k' \succeq k$ that $k' \Vdash A \vee B$, $k' \Vdash A \to C$ and $k' \Vdash B \to C$ (we can safely assume that k' is the same for all three premises.) Show that $k' \Vdash C$. By definition, there is an n s.t. for all $k'' \succeq_n k'$, $k'' \Vdash A$ or $k'' \Vdash B$. In both cases it follows that $k'' \Vdash C$, since $k' \Vdash A \to C$ and $k' \Vdash B \to C$. By the covering lemma, $k' \Vdash C$.

The cases $\wedge^+$, $\wedge^-$ are easy.

Case $\exists^+$: $A \to \exists_x A$. We show $k \Vdash (A \to \exists_x A)[\eta]$. Assume $k' \succeq k$ and $k' \Vdash A[\eta]$. We show $k' \Vdash (\exists_x A)[\eta]$. Since $\eta = \eta_x^{\eta(x)}$ there is an $a \in |\mathcal{T}|$ (namely $a := \eta(x)$) such that $k' \Vdash A[\eta_x^a]$. Hence, $k' \Vdash (\exists_x A)[\eta]$.

Case $\exists^-$: $\exists_x A \to \forall_x(A \to B) \to B$ and $x \notin \mathrm{FV}(B)$. We show that $k \Vdash (\exists_x A \to \forall_x(A \to B) \to B)[\eta]$. Assume that $k' \succeq k$ and $k' \Vdash (\exists_x A)[\eta]$ and $k' \Vdash \forall_x(A \to B)[\eta]$. We show $k' \Vdash B[\eta]$. By definition, there is an n such that for all $k'' \succeq_n k'$ we have $a \in |\mathcal{T}|$ and $k'' \Vdash A[\eta_x^a]$. From $k' \Vdash \forall_x(A \to B)[\eta]$ it follows that $k'' \Vdash B[\eta_x^a]$, and since $x \notin \mathrm{FV}(B)$, from the coincidence lemma, $k'' \Vdash B[\eta]$. Then, finally, by the covering lemma $k' \Vdash B[\eta]$.

This concludes the treatment of the axioms. We now consider the rules. In case of the assumption rule $u\colon A$ we have $A \in \Gamma$ and the claim is obvious.

Case $\to^+$. Assume $k \Vdash \Gamma$. We show $k \Vdash A \to B$. Assume $k' \succeq k$ and $k' \Vdash A$. Our goal is $k' \Vdash B$. We have $k' \Vdash \Gamma \cup \{A\}$. Thus, $k' \Vdash B$ by IH.

Case $\to^-$. Assume $k \Vdash \Gamma$. The IH gives us $k \Vdash A \to B$ and $k \Vdash A$. Hence $k \Vdash B$.

Case $\forall^+$. Assume $k \Vdash \Gamma[\eta]$ and $x \notin \mathrm{FV}(\Gamma)$. We show $k \Vdash (\forall_x A)[\eta]$, i.e., $k \Vdash A[\eta_x^a]$ for an arbitrary $a \in |\mathcal{T}|$. We have

$$
\begin{aligned}
&k \Vdash \Gamma[\eta_x^a] && \text{by the coincidence lemma, since } x \notin \mathrm{FV}(\Gamma)\\
&k \Vdash A[\eta_x^a] && \text{by IH.}
\end{aligned}
$$

Case $\forall^-$. Let $k \Vdash \Gamma[\eta]$. We show that $k \Vdash A(t)[\eta]$. This follows from

$$
\begin{aligned}
&k \Vdash (\forall_x A(x))[\eta] && \text{by IH}\\
&k \Vdash A(x)[\eta_x^{\eta(t)}] && \text{by definition}\\
&k \Vdash A(t)[\eta] && \text{by the substitution lemma.}
\end{aligned}
$$

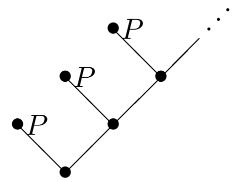
This concludes the proof. □

1.4.4. Counter models. With soundness at hand, it is easy to build counter models for derivations not valid in minimal or intuitionistic logic.

A *tree model for intuitionistic logic* is a tree model $\mathcal{T} = (D, I_0, I_1)$ in which $\bot$ is never forced, i.e., $I_1(\bot, k)$ is false for all k. Then

$$\begin{aligned} k \Vdash \neg A &\leftrightarrow \forall_{k' \succeq k}(k' \nVdash A), \\ k \Vdash \neg\neg A &\leftrightarrow \forall_{k' \succeq k}(k' \nVdash \neg A) \\ &\leftrightarrow \forall_{k' \succeq k} \tilde{\exists}_{k'' \succeq k'}(k'' \Vdash A). \end{aligned}$$

As an example we show that $\nvdash_i \neg\neg P \to P$. We describe the desired tree model by means of a diagram below. Next to every node, we write all propositions forced on that node.



Clearly this is an intuitionistic tree model, and $\langle\rangle \nVdash P$. Using the remark above, it is easily seen that $\langle\rangle \Vdash \neg\neg P$. Thus $\langle\rangle \nVdash (\neg\neg P \to P)$ and hence $\nVdash (\neg\neg P \to P)$. Since for every R and all k, $k \Vdash \forall_{\vec{x}}(\bot \to R\vec{x})$, it also follows that $\nvdash_i (\neg\neg P \to P)$. The model also shows that the *Peirce formula* $((P \to Q) \to P) \to P$ is invalid in intuitionistic logic.

1.4.5. Completeness.

THEOREM (Completeness). *Let* $\Gamma \cup \{A\}$ *be a set of formulas. Then the following propositions are equivalent.*

(a) $\Gamma \vdash A$.

(b) $\Gamma \Vdash A$, *i.e., for all tree models* $\mathcal{T}$, *nodes* k *and assignments* η

$$\mathcal{T}, k \Vdash \Gamma[\eta] \to \mathcal{T}, k \Vdash A[\eta].$$

PROOF. Soundness is one direction. For the other direction we employ a technique due to Harvey Friedman and construct a tree model $\mathcal{T}$ (over the set T_{01} of all finite 0-1-sequences k ordered by the initial segment relation $k \preceq k'$) with the property that $\Gamma \vdash B$ is equivalent to $\mathcal{T}, \langle\rangle \Vdash B[\mathrm{id}]$. We can assume here that Γ and also A are closed.

In order to define $\mathcal{T}$, we will need an enumeration $A_0, A_1, A_2, \dots$ of the underlying language $\mathcal{L}$, in which every formula occurs infinitely often. We also fix an enumeration $x_0, x_1, \dots$ of distinct variables. Write $\Gamma = \bigcup_n \Gamma_n$ with finite sets Γ_n such that $\Gamma_n \subseteq \Gamma_{n+1}$. With every node $k \in T_{01}$, we associate a finite set Δ_k of formulas and a set V_k of variables, by induction on the length of k.

Let $\Delta_{\langle\rangle} := \emptyset$ and $V_{\langle\rangle} := \emptyset$. Take a node k such that $\mathrm{lh}(k) = n$ and suppose that Δ_k, V_k are already defined. Write $\Delta \vdash_n B$ to mean that there is a derivation of length $\le n$ of B from Δ. We define Δ_{k0}, V_{k0} and Δ_{k1}, V_{k1} as follows:

Case 0. $\mathrm{FV}(A_n) \not\subseteq V_k$. Then let

$$\Delta_{k0} := \Delta_{k1} := \Delta_k \quad \text{and} \quad V_{k0} := V_{k1} := V_k.$$

Case 1. $\mathrm{FV}(A_n) \subseteq V_k$ and $\Gamma_n, \Delta_k \nvdash_n A_n$. Let

$$\Delta_{k0} := \Delta_k \quad \text{and} \quad \Delta_{k1} := \Delta_k \cup \{A_n\},$$
$$V_{k0} := V_{k1} := V_k.$$

Case 2. $\mathrm{FV}(A_n) \subseteq V_k$ and $\Gamma_n, \Delta_k \vdash_n A_n = A'_n \vee A''_n$. Let

$$\Delta_{k0} := \Delta_k \cup \{A_n, A'_n\} \quad \text{and} \quad \Delta_{k1} := \Delta_k \cup \{A_n, A''_n\},$$
$$V_{k0} := V_{k1} := V_k.$$

Case 3. $\mathrm{FV}(A_n) \subseteq V_k$ and $\Gamma_n, \Delta_k \vdash_n A_n = \exists_x A'_n(x)$. Let

$$\Delta_{k0} := \Delta_{k1} := \Delta_k \cup \{A_n, A'_n(x_i)\} \quad \text{and} \quad V_{k0} := V_{k1} := V_k \cup \{x_i\},$$

where x_i is the first variable $\notin V_k$.

Case 4. $\mathrm{FV}(A_n) \subseteq V_k$ and $\Gamma_n, \Delta_k \vdash_n A_n$, with A_n neither a disjunction nor an existentially quantified formula. Let

$$\Delta_{k0} := \Delta_{k1} := \Delta_k \cup \{A_n\} \quad \text{and} \quad V_{k0} := V_{k1} := V_k.$$

Obviously $\mathrm{FV}(\Delta_k) \subseteq V_k$, and $k \preceq k'$ implies that $\Delta_k \subseteq \Delta_{k'}$. Notice also that because of $\vdash \exists_x(\bot \to \bot)$ and this formula is repeated infinitely often in the given enumeration, for every variable x_i there is an m such that $x_i \in V_k$ for all k with $\mathrm{lh}(k) = m$.

We note that

$$\forall_{k' \succeq_n k}\,(\Gamma, \Delta_{k'} \vdash B) \to \Gamma, \Delta_k \vdash B, \quad \text{provided } \mathrm{FV}(B) \subseteq V_k. \tag{1.1}$$

It is sufficient to show that, for $\mathrm{FV}(B) \subseteq V_k$,

$$(\Gamma, \Delta_{k0} \vdash B) \wedge (\Gamma, \Delta_{k1} \vdash B) \to (\Gamma, \Delta_k \vdash B).$$

In cases 0, 1 and 4, this is obvious. For case 2, the claim follows immediately from the axiom scheme $\vee^-$. In case 3, we have $\mathrm{FV}(A_n) \subseteq V_k$ and $\Gamma_n, \Delta_k \vdash_n A_n = \exists_x A'_n(x)$. Assume $\Gamma, \Delta_k \cup \{A_n, A'_n(x_i)\} \vdash B$ with $x_i \notin V_k$, and $\mathrm{FV}(B) \subseteq V_k$. Then $x_i \notin \mathrm{FV}(\Delta_k \cup \{A_n, B\})$, hence $\Gamma, \Delta_k \cup \{A_n\} \vdash B$ by $\exists^-$ and therefore $\Gamma, \Delta_k \vdash B$.

Next, we show

$$\Gamma, \Delta_k \vdash B \to \exists_n \forall_{k' \succeq_n k}\,(B \in \Delta_{k'}), \quad \text{provided } \mathrm{FV}(B) \subseteq V_k. \tag{1.2}$$

Choose $n \ge \mathrm{lh}(k)$ such that $B = A_n$ and $\Gamma_n, \Delta_k \vdash_n A_n$. For all $k' \succeq k$, if $\mathrm{lh}(k') = n + 1$ then $A_n \in \Delta_{k'}$ (cf. the cases 2-4).

Using the sets Δ_k we can define a tree model $\mathcal{T}$ as (Ter, I_0, I_1) (where Ter denotes the set of terms of the underlying language $\mathcal{L}$) and the canonical $I_0(f)(\vec{s}) := f\vec{s}$ and

$$R^{\mathcal{T}}(\vec{s}, k) := R\vec{s} \in \Delta_k.$$

Obviously, $t^{\mathcal{T}}[\mathrm{id}] = t$ for all terms t.

Write $k \Vdash B$ for $\mathcal{T}, k \Vdash B[\mathrm{id}]$. We show that

$$\Gamma, \Delta_k \vdash B \leftrightarrow k \Vdash B, \quad \text{provided } \mathrm{FV}(B) \subseteq V_k. \tag{1.3}$$

The proof is by induction on B. *Case* $R\vec{s}$. Assume $\mathrm{FV}(R\vec{s}) \subseteq V_k$. The following are equivalent.

$\Gamma, \Delta_k \vdash R\vec{s}$	
$\exists_n \forall_{k' \succeq_n k} (R\vec{s} \in \Delta_{k'})$	by (1.2) and (1.1)
$\exists_n \forall_{k' \succeq_n k} R^{\mathcal{T}}(\vec{s}, k')$	by definition of $\mathcal{T}$
$k \Vdash R\vec{s}$	by definition of $\Vdash$, since $t^{\mathcal{T}}[\mathrm{id}] = t$.

Case $B \vee C$. Assume $\mathrm{FV}(B \vee C) \subseteq V_k$. $\rightarrow$. Let $\Gamma, \Delta_k \vdash B \vee C$. Choose an $n \geq \mathrm{lh}(k)$ such that $\Gamma_n, \Delta_k \vdash_n A_n = B \vee C$. Then, for all $k' \succeq k$ s.t. $\mathrm{lh}(k') = n$

$$\Delta_{k'0} = \Delta_{k'} \cup \{B \vee C, B\} \quad \text{and} \quad \Delta_{k'1} = \Delta_{k'} \cup \{B \vee C, C\},$$

and therefore by IH

$$k'0 \Vdash B \quad \text{and} \quad k'1 \Vdash C.$$

By definition, we have $k \Vdash B \vee C$. $\leftarrow$.

$k \Vdash B \vee C$	
$\exists_n \forall_{k' \succeq_n k} (k' \Vdash B \vee k' \Vdash C)$	
$\exists_n \forall_{k' \succeq_n k} ((\Gamma, \Delta_{k'} \vdash B) \vee (\Gamma, \Delta_{k'} \vdash C))$	by IH
$\exists_n \forall_{k' \succeq_n k} (\Gamma, \Delta_{k'} \vdash B \vee C)$	
$\Gamma, \Delta_k \vdash B \vee C$	by (1.1).

The case $B \wedge C$ is evident.

Case $B \rightarrow C$. Assume $\mathrm{FV}(B \rightarrow C) \subseteq V_k$. $\rightarrow$. Let $\Gamma, \Delta_k \vdash B \rightarrow C$. We must show $k \Vdash B \rightarrow C$, i.e.,

$$\forall_{k' \succeq k} (k' \Vdash B \rightarrow k' \Vdash C).$$

Let $k' \succeq k$ be such that $k' \Vdash B$. By IH, it follows that $\Gamma, \Delta_{k'} \vdash B$, and $\Gamma, \Delta_{k'} \vdash C$ follows by assumption. Then again by IH $k' \Vdash C$.

$\leftarrow$. Let $k \Vdash B \rightarrow C$, i.e., $\forall_{k' \succeq k} (k' \Vdash B \rightarrow k' \Vdash C)$. We show that $\Gamma, \Delta_k \vdash B \rightarrow C$, using (1.1). Choose $n \geq \mathrm{lh}(k)$ such that $B = A_n$. For all $k' \succeq_m k$ with $m := n - \mathrm{lh}(k)$ we show that $\Gamma, \Delta_{k'} \vdash B \rightarrow C$.

If $\Gamma, \Delta_{k'} \vdash_n A_n$, then $k' \Vdash B$ by IH, and $k' \Vdash C$ by assumption. Hence $\Gamma, \Delta_{k'} \vdash C$ again by IH and thus $\Gamma, \Delta_{k'} \vdash B \to C$.

If $\Gamma, \Delta_{k'} \nvdash_n A_n$, then by definition $\Delta_{k'1} = \Delta_{k'} \cup \{B\}$. Hence $\Gamma, \Delta_{k'1} \vdash B$, and thus $k'1 \Vdash B$ by IH. Now $k'1 \Vdash C$ by assumption, and finally $\Gamma, \Delta_{k'1} \vdash C$ by IH. From $\Delta_{k'1} = \Delta_{k'} \cup \{B\}$ it follows that $\Gamma, \Delta_{k'} \vdash B \to C$.

Case $\forall_x B(x)$. Assume $\mathrm{FV}(\forall_x B(x)) \subseteq V_k$. $\to$. Let $\Gamma, \Delta_k \vdash \forall_x B(x)$. Fix a term t. Then $\Gamma, \Delta_k \vdash B(t)$. Choose n such that $\mathrm{FV}(B(t)) \subseteq V_{k'}$ for all $k' \succeq_n k$. Then $\forall_{k' \succeq_n k} (\Gamma, \Delta_{k'} \vdash B(t))$, hence $\forall_{k' \succeq_n k} (k' \Vdash B(t))$ by IH, hence $k \Vdash B(t)$ by the covering lemma. This holds for every term t, hence $k \Vdash \forall_x B(x)$.

$\leftarrow$. Assume $k \Vdash \forall_x B(x)$. Pick $k' \succeq_n k$ such that $A_m = \exists_x(\bot \to \bot)$, for $m := \mathrm{lh}(k) + n$. Then at height m we put some x_i into the variable sets: for $k' \succeq_n k$ we have $x_i \notin V_{k'}$ but $x_i \in V_{k'j}$. Clearly $k'j \Vdash B(x_i)$, hence $\Gamma, \Delta_{k'j} \vdash B(x_i)$ by IH, hence (since at this height we consider the trivial formula $\exists_x(\bot \to \bot)$) also $\Gamma, \Delta_{k'} \vdash B(x_i)$. Since $x_i \notin V_{k'}$ we obtain $\Gamma, \Delta_{k'} \vdash \forall_x B(x)$. This holds for all $k' \succeq_n k$, hence $\Gamma, \Delta_k \vdash \forall_x B(x)$ by (1.1).

Case $\exists_x B(x)$. Assume $\mathrm{FV}(\exists_x B(x)) \subseteq V_k$. $\to$. Let $\Gamma, \Delta_k \vdash \exists_x B(x)$. Choose an $n \geq \mathrm{lh}(k)$ such that $\Gamma_n, \Delta_k \vdash_n A_n = \exists_x B(x)$. Then, for all $k' \succeq k$ with $\mathrm{lh}(k') = n$

$$\Delta_{k'0} = \Delta_{k'1} = \Delta_k \cup \{\exists_x B(x), B(x_i)\}$$

where $x_i \notin V_{k'}$. Hence by IH for $B(x_i)$ (applicable since $\mathrm{FV}(B(x_i)) \subseteq V_{k'j}$ for $j = 0, 1$)

$$k'0 \Vdash B(x_i) \quad \text{and} \quad k'1 \Vdash B(x_i).$$

It follows by definition that $k \Vdash \exists_x B(x)$.

$\leftarrow$. Assume $k \Vdash \exists_x B(x)$. Then $\forall_{k' \succeq_n k} \exists_{t \in \mathrm{Ter}} (k' \Vdash B(x)[\mathrm{id}_x^t])$ for some n, hence $\forall_{k' \succeq_n k} \exists_{t \in \mathrm{Ter}} (k' \Vdash B(t))$. For each of the finitely many $k' \succeq_n k$ pick an m such that $\forall_{k'' \succeq_m k'} (\mathrm{FV}(B(t_{k'})) \subseteq V_{k''})$. Let m_0 be the maximum of all these m. Then

$$\forall_{k'' \succeq_{m_0+n} k} \exists_{t \in \mathrm{Ter}} ((k'' \Vdash B(t)) \wedge \mathrm{FV}(B(t)) \subseteq V_{k''}).$$

The IH for $B(t)$ yields

$$\begin{aligned}
&\forall_{k'' \succeq_{m_0+n} k} \exists_{t \in \mathrm{Ter}} (\Gamma, \Delta_{k''} \vdash B(t)) \\
&\forall_{k'' \succeq_{m_0+n} k} (\Gamma, \Delta_{k''} \vdash \exists_x B(x)) \\
&\Gamma, \Delta_k \vdash \exists_x B(x) && \text{by (1.1).}
\end{aligned}$$

Now we can finish the proof of the completeness theorem. We apply (b) to the tree model $\mathcal{T}$ constructed above from Γ, the empty node $\langle\rangle$ and the assignment $\eta = \mathrm{id}$. Then $\mathcal{T}, \langle\rangle \Vdash \Gamma[\mathrm{id}]$ by (1.3), hence $\mathcal{T}, \langle\rangle \Vdash A[\mathrm{id}]$ by assumption and therefore $\Gamma \vdash A$ by (1.3) again. □

Completeness of intuitionistic logic follows as a corollary.

COROLLARY. *Let $\Gamma \cup \{A\}$ be a set of formulas. The following propositions are equivalent.*

(a) $\Gamma \vdash_i A$.

(b) $\Gamma, \mathrm{Efq} \Vdash A$, *i.e., for all tree models $\mathcal{T}$ for the intuitionistic logic, nodes k and assignments η*

$$\mathcal{T}, k \Vdash \Gamma[\eta] \to \mathcal{T}, k \Vdash A[\eta]. \qquad \square$$

1.5. Soundness and Completeness of the Classical Fragment

We give a proof of completeness of classical logic relying on the completeness proof for minimal logic above.

1.5.1. Models. We define the notion of a (classical) model (or more accurately, $\mathcal{L}$-model), and what the value of a term and the meaning of a formula in a model should be. The latter definition is by induction on formulas, where in the quantifier case we need a quantifier in the definition.

For the rest of this section, fix a countable formal language $\mathcal{L}$; we do not mention the dependence on $\mathcal{L}$ in the notation. Since we deal with classical logic, we only consider formulas built without $\vee, \exists$.

DEFINITION. A *model* is a triple $\mathcal{M} = (D, I_0, I_1)$ such that

(a) D is a nonempty set;

(b) for every n-ary function symbol f, I_0 assigns to f a map $I_0(f)\colon D^n \to D$;

(c) for every n-ary relation symbol R, I_1 assigns to R an n-ary relation on D^n. In case $n = 0$, $I_1(R)$ is either true or false. We require that $I_1(\bot)$ is false.

We write $|\mathcal{M}|$ for the carrier set D of $\mathcal{M}$ and $f^{\mathcal{M}}$, $R^{\mathcal{M}}$ for the interpretations $I_0(f)$, $I_1(R)$ of the function and relation symbols. *Assignments* η and their homomorphic extensions are defined as in 1.4.1. Again we write $t^{\mathcal{M}}[\eta]$ for $\eta(t)$.

DEFINITION (Validity). For every model $\mathcal{M}$, assignment η in $|\mathcal{M}|$ and formula A such that $\mathrm{FV}(A) \subseteq \mathrm{dom}(\eta)$ we define $\mathcal{M} \models A[\eta]$ (read: A is *valid* in $\mathcal{M}$ under the assignment η) by induction on A.

$$\begin{aligned}
\mathcal{M} &\models (R\vec{s})[\eta] &&:= R^{\mathcal{M}}(\vec{s}^{\mathcal{M}}[\eta]),\\
\mathcal{M} &\models (A \to B)[\eta] &&:= ((\mathcal{M} \models A[\eta]) \to (\mathcal{M} \models B[\eta])),\\
\mathcal{M} &\models (A \wedge B)[\eta] &&:= ((\mathcal{M} \models A[\eta]) \wedge (\mathcal{M} \models B[\eta])),\\
\mathcal{M} &\models (\forall_x A)[\eta] &&:= \forall_{a \in |\mathcal{M}|}(\mathcal{M} \models A[\eta_x^a]).
\end{aligned}$$

Since $I_1(\bot)$ is false, we have $\mathcal{M} \not\models \bot[\eta]$.

1.5.2. Soundness of classical logic.

LEMMA (Coincidence). *Let $\mathcal{M}$ be a model, t a term, A a formula and η, ξ assignments in $|\mathcal{M}|$.*

(a) *If $\eta(x) = \xi(x)$ for all $x \in \mathrm{vars}(t)$, then $\eta(t) = \xi(t)$.*
(b) *If $\eta(x) = \xi(x)$ for all $x \in \mathrm{FV}(A)$, then $\mathcal{M} \models A[\eta]$ if and only if $\mathcal{M} \models A[\xi]$.*

PROOF. Induction on terms and formulas. □

LEMMA (Substitution). *Let $\mathcal{M}$ be a model, t, r terms, A a formula and η an assignment in $|\mathcal{M}|$. Then*

(a) $\eta(r(t)) = \eta_x^{\eta(t)}(r(x))$.
(b) *$\mathcal{M} \models A(t)$ if and only if $\mathcal{M} \models A(x)[\eta_x^{\eta(t)}]$.*

PROOF. Induction on terms and formulas. □

A model $\mathcal{M}$ is called *classical* if $\neg\neg R^{\mathcal{M}}(\vec{a}\,) \to R^{\mathcal{M}}(\vec{a}\,)$ for all relation symbols R and all $\vec{a} \in |\mathcal{M}|$. We prove that every formula derivable in classical logic is valid in an arbitrary classical model.

THEOREM (Soundness of classical logic). *Let $\Gamma \cup \{A\}$ be a set of formulas such that $\Gamma \vdash_c A$. Then, if $\mathcal{M}$ is a classical model and η an assignment in $|\mathcal{M}|$, it follows that $\mathcal{M} \models \Gamma[\eta]$ entails $\mathcal{M} \models A[\eta]$.*

PROOF. Induction on derivations. We begin with the axioms in Stab and the axiom schemes $\wedge^+$, $\wedge^-$. $\mathcal{M} \models C[\eta]$ is abbreviated $\mathcal{M} \models C$ when η is known from the context.

For the stability axiom $\forall_{\vec{x}}(\neg\neg R\vec{x} \to R\vec{x}\,)$ the claim follows from our assumption that $\mathcal{M}$ is classical, i.e., $\neg\neg R^{\mathcal{M}}(\vec{a}\,) \to R^{\mathcal{M}}(\vec{a}\,)$ for all $\vec{a} \in |\mathcal{M}|$. The axioms $\wedge^+$, $\wedge^-$ are clearly valid.

This concludes the treatment of the axioms. We now consider the rules. In case of the assumption rule $u\colon A$ we have $A \in \Gamma$ and the claim is obvious.

Case $\to^+$. Assume $\mathcal{M} \models \Gamma$. We show $\mathcal{M} \models (A \to B)$. So assume in addition $\mathcal{M} \models A$. We must show $\mathcal{M} \models B$. By IH (with $\Gamma \cup \{A\}$ instead of Γ) this clearly holds.

Case $\to^-$. Assume $\mathcal{M} \models \Gamma$. We must show $\mathcal{M} \models B$. By IH, $\mathcal{M} \models (A \to B)$ and $\mathcal{M} \models A$. The claim follows from the definition of $\models$.

Case $\forall^+$. Assume $\mathcal{M} \models \Gamma[\eta]$ and $x \notin \mathrm{FV}(\Gamma)$. We show $\mathcal{M} \models (\forall_x A)[\eta]$, i.e., $\mathcal{M} \models A[\eta_x^a]$ for an arbitrary $a \in |\mathcal{M}|$. We have

$$\mathcal{M} \models \Gamma[\eta_x^a] \quad \text{by the coincidence lemma, since } x \notin \mathrm{FV}(\Gamma)$$
$$\mathcal{M} \models A[\eta_x^a] \quad \text{by IH.}$$

Case $\forall^-$. Let $\mathcal{M} \models \Gamma[\eta]$. We show that $\mathcal{M} \models A(t)[\eta]$. This follows from

$$\begin{aligned} &\mathcal{M} \models (\forall_x A(x))[\eta] && \text{by IH} \\ &\mathcal{M} \models A(x)[\eta_x^{\eta(t)}] && \text{by definition} \\ &\mathcal{M} \models A(t)[\eta] && \text{by the substitution lemma.} \end{aligned}$$

This concludes the proof. □

1.5.3. Completeness of classical logic. In our metatheory we use constructively valid arguments only, and mention explicitely all assumptions going beyond. A consequence is that when dealing with the classical fragment we need to restrict to classical models (satisfying $\neg\neg R^{\mathcal{M}}(\vec{a}\,) \to R^{\mathcal{M}}(\vec{a}\,)$ for all relation symbols R and all $\vec{a} \in |\mathcal{M}|$). In the proof of the completeness theorem below we will make use of the *axiom of dependent choice* for the weak existential quantifier

$$\tilde{\exists}_x A(0, x) \to \forall_{n,x}(A(n, x) \to \tilde{\exists}_y A(n + 1, y)) \to \tilde{\exists}_f \forall_n A(n, fn).$$

Recall that we only consider formulas without $\vee, \exists$.

THEOREM (Completeness of classical logic). *Let* $\Gamma \cup \{A\}$ *be a set of formulas. Assume that for all classical models* $\mathcal{M}$ *and assignments* η*,*

$$\mathcal{M} \models \Gamma[\eta] \to \mathcal{M} \models A[\eta].$$

Then, assuming the axiom of dependent choice for the weak existential quantifier, there must exist a derivation of A *from* $\Gamma \cup \mathrm{Stab}$.

PROOF. Since "there must exist a derivation" expresses the weak existential quantifier in the metalanguage, we need to prove a contradiction from the assumption $\Gamma, \mathrm{Stab} \nvdash A$.

By the completeness theorem for minimal logic, there must be a tree model $\mathcal{T} = (\mathrm{Ter}, I_0, I_1)$ on the complete binary tree T_{01} and a node l_0 such that $l_0 \Vdash \Gamma, \mathrm{Stab}$ and $l_0 \nVdash A$ (writing $k \Vdash B$ for $\mathcal{T}, k \Vdash B[\mathrm{id}]$).

Call a node k is *consistent* if $k \nVdash \bot$, and *stable* if $k \Vdash \mathrm{Stab}$. We prove

$$k \nVdash B \to \tilde{\exists}_{k' \succeq k}(k' \Vdash \neg B \wedge k \nVdash \bot) \qquad (k \text{ stable}). \tag{1.4}$$

Let k be a stable node, and B a formula (without $\vee$, $\exists$). Then, $\mathrm{Stab} \vdash \neg\neg B \to B$ by the stability lemma, and therefore $k \Vdash \neg\neg B \to B$. Hence from $k \nVdash B$ we obtain $k \nVdash \neg\neg B$. By a remark in 1.4.4 this implies that $\neg\forall_{k' \succeq k}(k' \Vdash \neg B \to k' \Vdash \bot)$, which proves (1.4).

Let α be a branch in the underlying tree T_{01}. We define

$$\begin{aligned} &\alpha \Vdash A && := \tilde{\exists}_{k \in \alpha}(k \Vdash A), \\ &\alpha \text{ is consistent} && := \alpha \nVdash \bot, \\ &\alpha \text{ is stable} && := \tilde{\exists}_{k \in \alpha}(k \Vdash \mathrm{Stab}). \end{aligned}$$

Note that from $\alpha \Vdash \vec{A}$ and $\vdash \vec{A} \to B$ it follows that $\alpha \Vdash B$. To see this, consider $\alpha \Vdash \vec{A}$. Then $k \Vdash \vec{A}$ for a $k \in \alpha$, since α is linearly ordered. From $\vdash \vec{A} \to B$ it follows that $k \Vdash B$, i.e., $\alpha \Vdash B$.

A branch α is *generic* (in the sense that it generates a classical model) if it is consistent and stable, if in addition for all formulas B

$$(\alpha \Vdash B) \tilde{\vee} (\alpha \Vdash \neg B), \tag{1.5}$$

and for all formulas $\forall_{\vec{y}} B(\vec{y})$ (with $\vec{y}$ possibly empty) where $B(\vec{y})$ is not a universal formula

$$\forall_{\vec{s} \in \mathrm{Ter}}(\alpha \Vdash B(\vec{s})) \to \alpha \Vdash \forall_{\vec{y}} B(\vec{y}). \tag{1.6}$$

For a branch α, we define a model $\mathcal{M}^\alpha = (\mathrm{Ter}, I_0, I_1^\alpha)$ as

$$I_1^\alpha(R)(\vec{s}) := \tilde{\exists}_{k \in \alpha} I_1(R, k)(\vec{s}) \qquad (R \neq \bot).$$

Since $\tilde{\exists}$ is used in this definition, $\mathcal{M}^\alpha$ is stable.

We show that for every generic branch α and formula B (without $\vee$, $\exists$)

$$\alpha \Vdash B \leftrightarrow \mathcal{M}^\alpha \models B. \tag{1.7}$$

The proof is by induction on the logical complexity of B.

Case $R\vec{s}$ with $R \neq \bot$. Then (1.7) holds for all α.

Case $\bot$. We have $\alpha \not\Vdash \bot$ for all consistent α.

Case $B \to C$. $\to$. Let $\alpha \Vdash B \to C$ and $\mathcal{M}^\alpha \models B$. We must show that $\mathcal{M}^\alpha \models C$. Note that $\alpha \Vdash B$ by IH, hence $\alpha \Vdash C$, hence $\mathcal{M}^\alpha \models C$ again by IH. $\leftarrow$. Let $\mathcal{M}^\alpha \models B \to C$. Notice that $\alpha \Vdash A$ is a negation; hence we can argue by a case distinction based on $\tilde{\vee}$. Clearly $(\mathcal{M}^\alpha \models B) \tilde{\vee} (\mathcal{M}^\alpha \not\models B)$. If $\mathcal{M}^\alpha \models B$, then $\mathcal{M}^\alpha \models C$. Hence $\alpha \Vdash C$ by IH and therefore $\alpha \Vdash B \to C$. If $\mathcal{M}^\alpha \not\models B$, then $\alpha \not\Vdash B$ by IH. Hence $\alpha \Vdash \neg B$ by (1.5) and therefore $\alpha \Vdash B \to C$, since α is stable (and $\vdash (\neg\neg C \to C) \to \bot \to C$).

Case $B \wedge C$. Easy.

Case $\forall_{\vec{y}} B(\vec{y})$ ($\vec{y}$ not empty) where $B(\vec{y})$ is not a universal formula. The following are equivalent.

$$\begin{array}{ll} \alpha \Vdash \forall_{\vec{y}} B(\vec{y}) & \\ \forall_{\vec{s} \in \mathrm{Ter}}(\alpha \Vdash B(\vec{s})) & \text{by (1.6)} \\ \forall_{\vec{s} \in \mathrm{Ter}}(\mathcal{M}^\alpha \models B(\vec{s})) & \text{by IH} \\ \mathcal{M}^\alpha \models \forall_{\vec{y}} B(\vec{y}). & \end{array}$$

This concludes the proof of (1.7)

Next we show that for every consistent and stable node k there must be a generic branch containing k:

$$k \not\Vdash \bot \to k \Vdash \mathrm{Stab} \to \tilde{\exists}_\alpha(\alpha \text{ generic} \wedge k \in \alpha). \tag{1.8}$$

For the proof, let $A_0, A_1, \dots$ enumerate all formulas. We define a sequence $k = k_0 \preceq k_1 \preceq k_2 \dots$ of consistent stable nodes by dependent choice. Let $k_0 := k$. Assume that k_n is defined. We write A_n in the form $\forall_{\vec{y}} B(\vec{y})$ (with $\vec{y}$ possibly empty) where B is not a universal formula. In case $k_n \Vdash \forall_{\vec{y}} B(\vec{y})$ let $k_{n+1} := k_n$. Otherwise we have $k_n \nVdash B(\vec{s})$ for some $\vec{s}$, and by (1.4) there must be a consistent node $k' \succeq k_n$ such that $k' \Vdash \neg B(\vec{s})$. Let $k_{n+1} := k'$. Since $k_n \preceq k_{n+1}$, the node k_{n+1} is stable.

Let $\alpha := \{ l \mid \exists_n (l \preceq k_n) \}$, hence $k \in \alpha$. We show that α is generic. Clearly α is consistent and stable. We now prove both (1.5) and (1.6). Let $C = \forall_{\vec{y}} B(\vec{y})$ (with $\vec{y}$ possibly empty) where $B(\vec{y})$ is not a universal formula, and choose n such that $C = A_n$. In case $k_n \Vdash \forall_{\vec{y}} B(\vec{y})$ we are done. Otherwise by construction $k_{n+1} \Vdash \neg B(\vec{s})$ for some $\vec{s}$. For (1.5) we get $k_{n+1} \Vdash \neg\forall_{\vec{y}} B(\vec{y})$ (since $\vdash \forall_{\vec{y}} B(\vec{y}) \to B(\vec{s})$), and (1.6) follows from the consistency of α. This concludes the proof of (1.8).

Now we can finalize the completeness proof. Recall that $l_0 \Vdash \Gamma, \mathrm{Stab}$ and $l_0 \nVdash A$. Since $l_0 \nVdash A$ and l_0 is stable, (1.4) yields a consistent node $k \succeq l_0$ such that $k \Vdash \neg A$. Evidently, k is stable as well. By (1.8) there must be a generic branch α such that $k \in \alpha$. Since $k \Vdash \neg A$ it follows that $\alpha \Vdash \neg A$, hence $\mathcal{M}^\alpha \models \neg A$ by (1.7). Moreover, $\alpha \Vdash \Gamma$, thus $\mathcal{M}^\alpha \models \Gamma$ by (1.7). This contradicts our assumption. □

1.5.4. Compactness theorem and the Löwenheim-Skolem theorem. From the many important corollaries of the completeness theorem we mention only two. A set Γ of formulas is *consistent* if $\Gamma \nvdash_c \bot$, and *satisfiable* if there is (in the weak sense) a classical model $\mathcal{M}$ and an assignment η in $|\mathcal{M}|$ such that $\mathcal{M} \models \Gamma[\eta]$.

Corollary. *Assume the axiom of dependent choice for the weak existential quantifier. Let Γ be a set of formulas.*

(a) *If Γ is consistent, then Γ is satisfiable.*

(b) *(Compactness). If each finite subset of Γ is satisfiable, Γ is satisfiable.*

Proof. (a). Assume $\Gamma \nvdash_c \bot$ and that for all classical models $\mathcal{M}$ we have $\mathcal{M} \not\models \Gamma$, i.e., $\mathcal{M} \models \Gamma$ implies $\mathcal{M} \models \bot$. Then the completeness theorem yields a contradiction.

(b). Otherwise by the completeness theorem there must be a derivation of $\bot$ from $\Gamma \cup \mathrm{Stab}$, hence also from $\Gamma_0 \cup \mathrm{Stab}$ for some finite subset $\Gamma_0 \subseteq \Gamma$. This contradicts the assumption that Γ_0 is satisfiable. □

Corollary (Löwenheim and Skolem). *Let Γ be a set of formulas (we assume that $\mathcal{L}$ is countable). If Γ is satisfiable, then Γ is satisfiable in a model with a countable carrier set.*