CHAPTER 1

Formal Proofs

The main subject of Mathematical Logic is mathematical proof. In this introductory chapter we deal with the basics of formalizing such proofs and, via normalization, analysing their structure. The system we pick for the representation of proofs is Gentzen's natural deduction from (1935). 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 proceed 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.

An underlying theme of this chapter is to bring out the constructive content of logic, particularly in regard to the relationship between minimal and classical logic. For us the latter is most appropriately viewed as a subsystem of the former. This approach will reveal some interesting aspects of proofs, e.g., that it is possible and useful to distinguish beween existential proofs that actually construct witnessing objects, and others that don't.

As an example for a non-constructive existence proof, consider the following proposition.

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

This can be proved as follows, by cases.

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

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

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

is rational.

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

Weyl (1921) gave a somewhat drastic description of this situation:

> Ein Existentialsatz – etwa "es gibt eine gerade Zahl" – ist überhaupt kein Urteil im eigentlichen Sinne, das einen Sachverhalt behauptet; Existentialsachverhalte sind eine leere Erfindung der Logiker. "2 ist eine gerade Zahl", das ist ein wirkliches, einem Sachverhalt Ausdruck gebendes Urteil; "es gibt eine gerade Zahl" ist nur ein aus diesem Urteil gewonnenes Urteilsabstrakt. Bezeichne ich Erkenntnis als einen wertvollen Schatz, so ist das Urteilsabstrakt ein Papier, welches das Vorhandensein eines Schatzes anzeigt, ohne jedoch zu verraten, an welchem Ort. Sein einziger Wert kann darin liegen, daß es mich antreibt, nach dem Schatze zu suchen.

1.1. Natural deduction

The rules of natural deduction come in pairs: we have an introduction and an elimination rule for each of the logical connectives. The resulting system is called *minimal logic*; it was introduced by Kolmogorov (1932), Gentzen (1935) and Johansson (1937). Notice that no negation is yet present. If we go on and require *ex-falso-quodlibet* for the nullary propositional symbol $\bot$ ("falsum") we can embed *intuitionistic logic* with negation as $A \to \bot$. To embed classical logic, we need to go further and add as an axiom schema the principle of *indirect proof*, also called *stability* ($\forall_{\vec{x}}(\neg\neg R\vec{x} \to R\vec{x})$ for relation symbols R), but then it is appropriate to restrict to the language based on $\to$, $\forall$, $\bot$ and $\wedge$. The reason for this restriction is that we can neither prove $\neg\neg\exists_x A \to \exists_x A$ nor $\neg\neg(A \vee B) \to A \vee B$, for there are countermodels to both (the former is Markov's scheme). However, we can prove them for the classical existential quantifier and disjunction defined by $\neg\forall_x\neg A$ and $\neg A \to \neg B \to \bot$. Thus we need to make a distinction between two kinds of "exists" and two kinds of "or": the classical ones are "weak" and the non-classical ones "strong" since they have constructive content. In situations where both kinds occur together we must mark the distinction, and we shall do this by writing a tilde above the weak disjunction and existence symbols thus $\tilde{\vee}, \tilde{\exists}$.

1.1.1. Terms and formulas. Let a countably 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.

(i) 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.

(ii) 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 symbols can again 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.

(i) Every variable is a term.

(ii) Every constant is a term.

(iii) If $t_1, \ldots, t_n$ are terms and f is an n-ary function symbol with $n \geq 1$, then $f(t_1, \ldots, t_n)$ is a term. (If t, s are terms and $\circ$ is a binary function symbol, then $(t \circ s)$ is a term.)

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

Formulas are inductively defined from prime formulas by

(i) Every prime formula is a formula.

(ii) 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").

(iii) If A is a formula and x is a variable, then $\forall_x A$ ("A holds for all x") and $\exists_x A$ ("there is an x such that A") are formulas.

Negation is defined by

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

We shall often need to do induction on the height, denoted $|A|$, of formulas A. This is defined as follows: $|P| = 0$ for atoms P, $|A \circ B| = \max(|A|, |B|) + 1$ for binary operators $\circ$ (i.e., $\to, \wedge, \vee$) and $|\circ A| = |A| + 1$ for unary operators $\circ$ (i.e., $\forall_x$, $\exists_x$).

1.1.2. Substitution, free and bound variables. Expressions $\mathcal{E}, \mathcal{E}'$ which differ only in the names of bound (occurrences of) variables will be regarded as identical. This is sometimes expressed by saying that $\mathcal{E}$ and $\mathcal{E}'$ are α-equal. In other words, we are only interested in expressions "modulo renaming of bound variables". There are methods of finding unique representatives for such expressions, e.g., 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. A formula A is said to be *closed* if $\mathrm{FV}(A) = \emptyset$.

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

In a given context we shall adopt the following convention. 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.

1.1.3. Subformulas. Unless stated otherwise, the notion of *subformula* will be that defined by Gentzen.

DEFINITION. (Gentzen) subformulas of A are defined by

(a) A is a subformula of A;
(b) if $B \circ C$ is a subformula of A then so are B, C, for $\circ = \rightarrow, \wedge, \vee$;
(c) if $\forall_x B(x)$ or $\exists_x B(x)$ is a subformula of A, then so is $B(t)$.

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 $B \rightarrow C$ is a positive (negative) subformula of A, then B is a negative (positive) subformula of A, and C is a positive (negative) subformula of A;
(e) if $B \rightarrow C$ is a strictly positive subformula of A, then so is C.

A strictly positive subformula of A is also called a *strictly positive part (s.p.p.)* of A. Note that the set of subformulas of A is the union of the positive and negative subformulas of A.

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.

1.1.4. Examples of derivations. To motivate the rules for natural deduction, let us start with informal proofs of some simple logical facts.

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

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. Using the third assumption twice we have $B \to C$ by the first assumption, and B by the second assumption. From $B \to C$ and B we then obtain C. Then $A \to C$, cancelling the assumption on A; $(A \to B) \to A \to C$ cancelling the second assumption; and the result follows by cancelling the first assumption. □

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

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. Hence $\forall_x B$. Hence $A \to \forall_x B$, cancelling the second assumption. Hence the result, cancelling the first assumption. □

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 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 each node of which denotes a rule application. The labels of the inner nodes are the formulas derived as conclusions at those points, and the labels of the leaves are formulas or terms. The labels of the nodes immediately above a node k are the *premises* of the rule application. At the root of the tree we have the conclusion (or end formula) of the whole derivation. In natural deduction systems one works with *assumptions* at leaves of the tree; they can be either *open* or *closed* (cancelled). Any of these assumptions carries a *marker*. As markers we use *assumption variables* denoted $u, v, w, u_0, u_1, \ldots$. The variables of the language previously introduced will

now often be called *object variables*, to distinguish them from assumption variables. If at a node below an assumption the dependency on this assumption is removed (it becomes closed) we record this by writing down the assumption variable. Since the same assumption may be used more than once (this was the case in the first example above), the assumption marked with u (written $u\colon A$) may appear many times. Of course we insist that distinct assumption formulas must have distinct markers. An inner node of the tree is understood as the result of passing from premises to the conclusion of 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 or cancels an assumption variable u (and hence removes the dependency of all assumptions $u\colon A$ thus marked). An application of the $\forall$-introduction rule similarly binds an object variable x (and hence removes the dependency on x). In both cases the bound assumption or object variable is added to the label of the node.

Definition. A formula A is called *derivable* (in *minimal logic*), written $\vdash A$, if there is a derivation of A (without free assumptions) using the natural deduction rules. 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 $A_1, \dots, A_n \in \Gamma$.

We now formulate the rules of natural deduction.

1.1.5. Introduction and elimination rules for $\to$ and $\forall$. First we have an assumption rule, allowing to write down an arbitrary formula A together with a marker u:

$$u\colon A \qquad \text{assumption.}$$

The other rules of natural deduction split into introduction rules (I-rules for short) and elimination rules (E-rules) for the logical connectives which, for the time being, are just $\to$ and $\forall$. For implication $\to$ there is an introduction rule $\to^+$ and an elimination rule $\to^-$ also called *modus ponens*. The left premise $A \to B$ in $\to^-$ is called the *major* (or *main*) premise, and the right premise A the *minor* (or *side*) premise. Note that with an application of the $\to^+$-rule *all* assumptions above it marked with $u\colon A$ are cancelled (which is denoted by putting square brackets around these assumptions), and the u then gets written alongside. There may of course be other uncancelled assumptions $v\colon A$ of the same formula A, which may get cancelled at a later

stage.

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

For the universal quantifier $\forall$ there is an introduction rule $\forall^+$ (again marked, but now with the bound variable x) and an elimination rule $\forall^-$ whose right premise is the term t to be substituted. The rule $\forall^+ x$ with conclusion $\forall_x A$ is subject to the following *(eigen-)variable condition*: the derivation M of the premise A must not contain any open assumption having x as a free variable.

$$\frac{\begin{array}{c}\mid M\\ A\end{array}}{\forall_x A}\,\forall^{+}x \qquad \frac{\begin{array}{cc}\mid M & \\ \forall_x A(x) & t\end{array}}{A(t)}\,\forall^{-}$$

We now give derivations of the two example formulas treated informally above. Since in many cases the rule used is determined by the conclusion, we suppress in such cases the name of the rule.

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

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

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

1.1.6. Properties of negation. 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 (without stability – we will come back to this later on).

LEMMA. *The following are derivable.*

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

Derivations are left as an exercise.

1.1.7. Introduction and elimination rules for disjunction $\vee$, conjunction $\wedge$ and existence $\exists$. For disjunction the introduction and elimination rules are

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

For conjunction we have

$$\frac{\begin{array}{cc}|\,M & |\,N\\ A & B\end{array}}{A \wedge B}\wedge^+ \qquad \frac{\begin{array}{ccc} & [u\colon A] & [v\colon B]\\ |\,M & & |\,N\\ A \wedge B & & C\end{array}}{C}\wedge^- u, v$$

and for the existential quantifier

$$\frac{\begin{array}{cc} & |\,M\\ t & A(t)\end{array}}{\exists_x A(x)}\exists^+ \qquad \frac{\begin{array}{ccc} & [u\colon A] & \\ |\,M & & |\,N\\ \exists_x A & & B\end{array}}{B}\exists^- x, u \text{ (var.cond.)}$$

Similar to $\forall^+ x$ the rule $\exists^- x, u$ is subject to an *(eigen-)variable condition*: in the derivation N the variable x (i) should not occur free in the formula of any open assumption other than $u\colon A$, and (ii) should not occur free in B.

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

It is easy to see that for each of the connectives $\vee$, $\wedge$, $\exists$ the rules and the following axioms are equivalent over minimal logic; this is left as an exercise.

For disjunction the introduction and elimination axioms are

$$\begin{aligned}
&\vee_0^+ \colon A \to A \vee B,\\
&\vee_1^+ \colon B \to A \vee B,\\
&\vee^- \colon A \vee B \to (A \to C) \to (B \to C) \to C.
\end{aligned}$$

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 schema, that of an *inductively defined predicate*, which is defined by some introduction rules and one elimination rule.

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

Lemma. *The following are derivable.*

$$\begin{aligned}
&(A \wedge B \to C) \leftrightarrow (A \to B \to C),\\
&(A \to B \wedge C) \leftrightarrow (A \to B) \wedge (A \to C),\\
&\\
&(A \vee B \to C) \leftrightarrow (A \to C) \wedge (B \to C),\\
&(A \to B \vee C) \leftarrow (A \to B) \vee (A \to C),\\
&\\
&(\forall_x A \to B) \leftarrow \exists_x (A \to B) \quad \textit{if } x \notin \mathrm{FV}(B),\\
&(A \to \forall_x B) \leftrightarrow \forall_x (A \to B) \quad \textit{if } x \notin \mathrm{FV}(A),\\
&\\
&(\exists_x A \to B) \leftrightarrow \forall_x (A \to B) \quad \textit{if } x \notin \mathrm{FV}(B),\\
&(A \to \exists_x B) \leftarrow \exists_x (A \to B) \quad \textit{if } x \notin \mathrm{FV}(A).
\end{aligned}$$

Proof. A derivation of the final formula is

$$\dfrac{\dfrac{u \colon \exists_x (A \to B) \qquad \dfrac{\dfrac{\dfrac{}{x} \qquad \dfrac{w \colon A \to B \qquad v \colon A}{B}}{\exists_x B}\, \exists^- x, w}{\exists_x B}}{A \to \exists_x B}\, \to^+ v}{\exists_x (A \to B) \to A \to \exists_x B}\, \to^+ u$$

The variable condition for $\exists^-$ is satisfied since the variable x (i) is not free in the formula A of the open assumption $v \colon A$, and (ii) is not free in $\exists_x B$. The rest of the proof is left as an exercise. □

As already mentioned, we distinguish between two kinds of "exists" and two kinds of "or": the "weak" or classical ones and the "strong" or non-classical ones, with constructive content. In the present context both kinds occur together and hence we must mark the distinction; we shall do this by writing a tilde above the weak disjunction and existence symbols thus

$$A \mathbin{\tilde{\vee}} B := \neg A \to \neg B \to \bot, \qquad \tilde{\exists}_x A := \neg \forall_x \neg A.$$

These weak variants of disjunction and the existential quantifier are no stronger than the proper ones (in fact, they are weaker):

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

This can be seen easily by putting $C := \bot$ in $\vee^-$ and $B := \bot$ in $\exists^-$.

Remark. Since $\tilde{\exists}_x \tilde{\exists}_y A$ unfolds into a rather awkward formula we extend the $\tilde{\exists}$-terminology to lists of variables:

$$\tilde{\exists}_{x_1,\dots,x_n} A := \forall_{x_1,\dots,x_n}(A \to \bot) \to \bot.$$

Moreover let

$$\tilde{\exists}_{x_1,\dots,x_n}(A_1 \mathbin{\tilde{\wedge}} \dots \mathbin{\tilde{\wedge}} A_m) := \forall_{x_1,\dots,x_n}(A_1 \to \dots \to A_m \to \bot) \to \bot.$$

This allows to stay in the $\to, \forall$ part of the language. Notice that $\tilde{\wedge}$ only makes sense in this context, i.e., in connection with $\tilde{\exists}$.

1.1.8. Intuitionistic and classical derivability. In the definition of derivability in Section 1.1.4 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 *stability* 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).

(a) $\vdash (\neg\neg A \to A) \to (\neg\neg B \to B) \to \neg\neg(A \wedge B) \to A \wedge B$.
(b) $\vdash (\neg\neg B \to B) \to \neg\neg(A \to B) \to A \to B$.
(c) $\vdash (\neg\neg A \to A) \to \neg\neg\forall_x A \to A$.
(d) $\vdash_c \neg\neg A \to A$ *for every formula* A *without* $\vee, \exists$.

PROOF. (a) is left as an exercise.

(b) For simplicity, in the derivation to be constructed we leave out applications of $\to^+$ at the end.

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

(c)

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

(d) Induction on A. The case $R\vec{t}$ with R distinct from $\bot$ is given by Stab. In the case $\bot$ the desired derivation is

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

In the cases $A \wedge B$, $A \to B$ and $\forall_x A$ use (a), (b) and (c), respectively. □

Using stability we can prove some well-known facts about the interaction of weak disjunction and the weak existential quantifier with implication. We first prove a more refined claim, stating to what extent we need to go beyond minimal logic.

LEMMA. *The following are derivable.*

(1) $(\tilde{\exists}_x A \to B) \to \forall_x(A \to B)$ *if* $x \notin \mathrm{FV}(B)$,

(2) $(\neg\neg B \to B) \to \forall_x(A \to B) \to \tilde{\exists}_x A \to B$ *if* $x \notin \mathrm{FV}(B)$,

(3) $(\bot \to B[x{:=}c]) \to (A \to \tilde{\exists}_x B) \to \tilde{\exists}_x(A \to B)$ *if* $x \notin \mathrm{FV}(A)$,

(4) $\tilde{\exists}_x(A \to B) \to A \to \tilde{\exists}_x B$ *if* $x \notin \mathrm{FV}(A)$.

The last two items can also be seen as simplifying a weakly existentially quantified implication whose premise does not contain the quantified variable. In case the conclusion does not contain the quantified variable we have

(5) $(\neg\neg B \to B) \to \quad \tilde{\exists}_x(A \to B) \to \forall_x A \to B \qquad$ *if* $x \notin \mathrm{FV}(B)$,

(6) $\forall_x(\neg\neg A \to A) \to (\forall_x A \to B) \to \tilde{\exists}_x(A \to B) \quad$ *if* $x \notin \mathrm{FV}(B)$.

Proof. (1)

$$\dfrac{\tilde{\exists}_x A \to B \qquad \dfrac{\dfrac{u_1 \colon \forall_x \neg A \qquad x}{\neg A} \qquad A}{\bot}}{\neg\forall_x \neg A} \to^+ u_1}{B}$$

(2)

$$\dfrac{\neg\neg B \to B \qquad \dfrac{\dfrac{\neg\forall_x \neg A \qquad \dfrac{\dfrac{\dfrac{\forall_x(A \to B) \qquad x}{A \to B} \qquad u_1 \colon A}{B}}{\bot}}{\neg A}}{\forall_x \neg A}}{\bot} \to^+ u_1}{\neg\neg B} \to^+ u_2}{B}$$

(3) Writing B_0 for $B[x{:=}c]$ we have

$$\dfrac{\dfrac{\forall_x \neg(A \to B) \quad c}{\neg(A \to B_0)} \qquad \dfrac{\dfrac{\bot \to B_0 \qquad \dfrac{\dfrac{A \to \tilde{\exists}_x B \quad u_2 \colon A}{\tilde{\exists}_x B} \qquad \dfrac{\dfrac{\dfrac{\forall_x \neg(A \to B) \quad x}{\neg(A \to B)} \qquad \dfrac{u_1 \colon B}{A \to B}}{\bot}}{\neg B}}{\forall_x \neg B}}{\bot}}{B_0}}{A \to B_0} \to^+ u_2}{\bot}$$

(4)

$$\dfrac{\tilde{\exists}_x(A \to B) \qquad \dfrac{\dfrac{\dfrac{\forall_x \neg B \qquad x}{\neg B} \qquad \dfrac{u_1 \colon A \to B \qquad A}{B}}{\bot}}{\neg(A \to B)}}{\forall_x \neg(A \to B)} \to^+ u_1}{\bot}$$

(5)

$$\dfrac{\neg\neg B \to B \qquad \dfrac{\tilde{\exists}_x(A \to B) \qquad \dfrac{\dfrac{\dfrac{u_2\colon \neg B \qquad \dfrac{u_1\colon A \to B \qquad \dfrac{\forall_x A \quad x}{A}}{B}}{\bot}\to^+ u_1}{\neg(A \to B)}}{\forall_x \neg(A \to B)}}{\bot}\to^+ u_2}{\neg\neg B}}{B}$$

(6) We derive $\forall_x(\bot \to A) \to (\forall_x A \to B) \to \forall_x \neg(A \to B) \to \neg\neg A$. Writing Ax, Ay for $A(x), A(y)$ we have

$$\dfrac{\dfrac{\forall_x \neg(Ax \to B) \quad x}{\neg(Ax \to B)} \qquad \dfrac{\forall_x Ax \to B \qquad \dfrac{\dfrac{\dfrac{\forall_y(\bot \to Ay) \quad y}{\bot \to Ay} \qquad \dfrac{u_1\colon \neg Ax \quad u_2\colon Ax}{\bot}}{Ay}}{\forall_y Ay}}{B}}{Ax \to B}\to^+ u_2}{\bot}\to^+ u_1}{\neg\neg Ax}$$

Using this derivation M we obtain

$$\dfrac{\dfrac{\forall_x \neg(Ax \to B) \quad c}{\neg(Ac \to B)} \qquad \dfrac{\forall_x Ax \to B \qquad \dfrac{\dfrac{\dfrac{\forall_x(\neg\neg Ax \to Ax) \quad x}{\neg\neg Ax \to Ax} \qquad \neg\neg Ax}{Ax}}{\forall_x Ax}}{B}}{Ac \to B}}{\bot} \qquad \mid M$$

Since clearly $\vdash (\neg\neg A \to A) \to \bot \to A$ the claim follows. □

Remark. An immediate consequence of (6) is the classical derivability of the "drinker formula" $\tilde{\exists}_x(Px \to \forall_x Px)$, to be read "in every non-empty bar there is a person such that, if this person drinks, then everybody drinks". To see this let $A := Px$ and $B := \forall_x Px$ in (6). Note that we assumed that the language contains a constant c.

Corollary.

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

There is a similar lemma on weak disjunction:

Lemma. *The following are derivable.*

$$(A \mathbin{\tilde{\vee}} B \to C) \to (A \to C) \wedge (B \to C), \tag{7}$$
$$(\neg\neg C \to C) \to (A \to C) \to (B \to C) \to A \mathbin{\tilde{\vee}} B \to C, \tag{8}$$
$$(\bot \to B) \to \quad (A \to B \mathbin{\tilde{\vee}} C) \to (A \to B) \mathbin{\tilde{\vee}} (A \to C), \tag{9}$$
$$(A \to B) \mathbin{\tilde{\vee}} (A \to C) \to A \to B \mathbin{\tilde{\vee}} C, \tag{10}$$
$$(\neg\neg C \to C) \to (A \to C) \mathbin{\tilde{\vee}} (B \to C) \to A \to B \to C, \tag{11}$$
$$(\bot \to C) \to \quad (A \to B \to C) \to (A \to C) \mathbin{\tilde{\vee}} (B \to C). \tag{12}$$

Proof. We only consider (8) and (12); the rest is left as an exercise.
(8)

$$\dfrac{\dfrac{\neg\neg C \to C \qquad \dfrac{\dfrac{\dfrac{\neg A \to \neg B \to \bot \qquad \dfrac{\dfrac{A \to C \quad u_2 : A}{C}}{\bot}\,{\scriptstyle \to^+ u_2}}{\neg B \to \bot} \qquad \dfrac{\dfrac{B \to C \quad u_3 : B}{C}}{\bot}\,{\scriptstyle \to^+ u_3}}{\neg B}}{\bot}\,{\scriptstyle \to^+ u_1}}{\neg\neg C}}{C}$$

(12)

$$\dfrac{\dfrac{\neg(B \to C) \qquad \dfrac{\dfrac{\bot \to C \qquad \dfrac{\dfrac{\neg(A \to C) \qquad \dfrac{\dfrac{A \to B \to C \quad u_1 : A}{B \to C} \qquad u_2 : B}{C}}{A \to C}\,{\scriptstyle \to^+ u_1}}{\bot}}{C}}{B \to C}\,{\scriptstyle \to^+ u_2}}{\bot}$$

The general idea here is to view $\tilde{\vee}$ as a finitary version of $\tilde{\exists}$. □

COROLLARY.

$$\vdash_c (A \mathbin{\tilde\vee} B \to C) \leftrightarrow (A \to C) \wedge (B \to C) \quad \textit{for } C \textit{ without } \vee, \exists,$$
$$\vdash_i (A \to B \mathbin{\tilde\vee} C) \leftrightarrow (A \to B) \mathbin{\tilde\vee} (A \to C),$$
$$\vdash_c (A \to C) \mathbin{\tilde\vee} (B \to C) \leftrightarrow (A \to B \to C) \quad \textit{for } C \textit{ without } \vee, \exists.$$

The weak existential quantifier $\tilde\exists$ and weak disjunction $\tilde\vee$ satisfy the same introduction axioms as the strong ones: this follows from the derivability of $\exists_x A \to \tilde\exists_x A$ and $A \vee B \to A \mathbin{\tilde\vee} B$ (see Section 1.1.7). They also satisfy the same elimination axioms, provided one restricts the conclusion to stable formulas. For $\tilde\exists$ this has been proved in (2), and for $\tilde\vee$ in (8).

Therefore when proving a stable goal in minimal logic more proof techniques are available than in the general case. For instance, case distinction on an arbitrary formula A is possible by (8), since $A \mathbin{\tilde\vee} \neg A$ is (easily) derivable. Another important example is

LEMMA. *The following is derivable.*

$$\forall_x \neg A \mathbin{\tilde\vee} \tilde\exists_x A.$$

PROOF. Unfolding $\tilde\vee$ and $\tilde\exists$ gives

$$(\forall_x(A \to \bot) \to \bot) \to (\underbrace{(\forall_x(A \to \bot) \to \bot)}_{\tilde\exists_x A} \to \bot) \to \bot. \qquad \square$$

It is often helpful to use this lemma in a slightly more general form, for instancce

$$\forall_{x,y}(A \to B \to \bot) \mathbin{\tilde\vee} \tilde\exists_{x,y}(A \mathbin{\tilde\wedge} B).$$

The proof is again immediate, since the right hand side $\tilde\exists_{x,y}(A \mathbin{\tilde\wedge} B)$ unfolds into the negated left hand side.

1.1.9. Gentzen translation. Classical derivability $\Gamma \vdash_c B$ was defined in Section 1.1.8 by $\Gamma \cup \mathrm{Stab} \vdash B$. This embedding of classical logic into minimal logic can be expressed in a somewhat different and very explicit form, namely as a syntactic translation $A \mapsto A^g$ of formulas such that A is derivable in classical logic if and only if its translation A^g is derivable in minimal logic.

DEFINITION (Gentzen translation A^g).

$$\begin{aligned}
(R\vec{t})^g &:= \neg\neg R\vec{t} \quad \text{for } R \text{ distinct from } \bot,\\
\bot^g &:= \bot,\\
(A \vee B)^g &:= A^g \mathbin{\tilde{\vee}} B^g,\\
(\exists_x A)^g &:= \tilde{\exists}_x A^g,\\
(A \circ B)^g &:= A^g \circ B^g \quad \text{for } \circ = \to, \wedge,\\
(\forall_x A)^g &:= \forall_x A^g.
\end{aligned}$$

LEMMA (Stability of A^g). $\vdash \neg\neg A^g \to A^g$.

PROOF. Induction on A.

Case $R\vec{t}$ with R distinct from $\bot$. We must show $\neg\neg\neg\neg R\vec{t} \to \neg\neg R\vec{t}$, which is a special case of $\vdash \neg\neg\neg B \to \neg B$.

Case $\bot$. Use $\vdash \neg\neg\bot \to \bot$.

Case $A \vee B$. We must show $\vdash \neg\neg(A^g \mathbin{\tilde{\vee}} B^g) \to A^g \mathbin{\tilde{\vee}} B^g$, which is a special case of $\vdash \neg\neg(\neg C \to \neg D \to \bot) \to \neg C \to \neg D \to \bot$:

$$\dfrac{\neg\neg(\neg C \to \neg D \to \bot) \qquad \dfrac{\dfrac{\dfrac{u_1 \colon \neg C \to \neg D \to \bot \qquad \neg C}{\neg D \to \bot} \qquad \neg D}{\bot}}{\neg(\neg C \to \neg D \to \bot)} \to^+ u_1}{\bot}$$

Case $\exists_x A$. In this case we must show $\vdash \neg\neg\tilde{\exists}_x A^g \to \tilde{\exists}_x A^g$, but this is a special case of $\vdash \neg\neg\neg B \to \neg B$, because $\tilde{\exists}_x A^g$ is the negation $\neg\forall_x \neg A^g$.

Case $A \wedge B$. We must show $\vdash \neg\neg(A^g \wedge B^g) \to A^g \wedge B^g$. By induction hypothesis $\vdash \neg\neg A^g \to A^g$ and $\vdash \neg\neg B^g \to B^g$. Now use part (a) of the stability theorem in Section 1.1.8.

The cases $A \to B$ and $\forall_x A$ are similar, using parts (b) and (c) of the stability theorem instead. □

THEOREM. (a) $\Gamma \vdash_c A$ *implies* $\Gamma^g \vdash A^g$.
(b) $\Gamma^g \vdash A^g$ *implies* $\Gamma \vdash_c A$ *for* Γ, A *without* $\vee, \exists$.

PROOF. (a) We use induction on $\Gamma \vdash_c A$. In case of a stability axiom $\forall_{\vec{x}}(\neg\neg R\vec{x} \to R\vec{x})$ we must derive $\forall_{\vec{x}}(\neg\neg\neg\neg R\vec{x} \to \neg\neg R\vec{x})$, which is easy (as above). For the rules $\to^+$, $\to^-$, $\forall^+$, $\forall^-$, $\wedge^+$ and $\wedge^-$ the claim follows immediately from the induction hypothesis, using the same rule again. This works because the Gentzen translation acts as a homomorphism for these connectives. For the rules $\vee_i^+$, $\vee^-$, $\exists^+$ and $\exists^-$ the claim follows from the induction hypothesis and the remark on the elimination rules for $\tilde{\vee}, \tilde{\exists}$ in

Section 1.1.8. For example, in case $\exists^-$ the induction hypothesis gives

$$\begin{array}{c} \\ \mid M \\ \tilde{\exists}_x A^g \end{array} \quad \text{and} \quad \begin{array}{c} u\colon A^g \\ \mid N \\ B^g \end{array}$$

with $x \notin \mathrm{FV}(B^g)$. Now use $\vdash (\neg\neg B^g \to B^g) \to \tilde{\exists}_x A^g \to \forall_x(A^g \to B^g) \to B^g$. Its premise $\neg\neg B^g \to B^g$ is derivable by the lemma above.

(b) First note that $\vdash_c (B \leftrightarrow B^g)$ if B is without $\vee, \exists$. Now assume that Γ, A are without $\vee, \exists$. From $\Gamma^g \vdash A^g$ we obtain $\Gamma \vdash_c A$ as follows. We argue informally. Assume Γ. Then Γ^g by the note, hence A^g because of $\Gamma^g \vdash A^g$, hence A again by the note. □

1.2. Normal derivations

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 use "conversions" to remove such "local maxima" of complexity, thus reducing any given derivation to normal form. We also analyse the shape of derivations in normal form, and prove the *subformula property*, which says that every formula in a normal derivation is a subformula of the end-formula or else of an assumption. For simplicity we consider derivations involving $\to, \forall$-rules only.

1.2.1. The Curry-Howard correspondence. Since natural deduction derivations can be notationally cumbersome, it will be convenient to represent them as typed "derivation terms", where the derived formula is the "type" of the term (and displayed as a superscript). This representation goes under the name of *Curry-Howard correspondence*. It dates back to Curry (1930) and somewhat later Howard, published only in (1980), who noted that the types of the combinators used in combinatory logic are exactly the Hilbert style axioms for minimal propositional logic. Subsequently Martin-Löf (1972) transferred these ideas to a natural deduction setting where natural deduction proofs of formulas A now correspond exactly to lambda terms with type A. This representation of natural deduction proofs will henceforth be used consistently.

We give an inductive definition of such derivation terms for the $\to, \forall$-rules in Table 1 where for clarity we have written the corresponding derivations to the left. This can be extended to the rules for $\vee$, $\wedge$ and $\exists$, but we will not do this here.

Every derivation term carries a formula as its type. However, we shall usually leave these formulas implicit and write derivation terms without them. Note that every derivation term can be written uniquely in one of

Derivation	Term
$u\colon A$	u^A
$\dfrac{\begin{array}{c}[u\colon A]\\ \mid M\\ B\end{array}}{A \to B} \to^+ u$	$(\lambda_{u^A} M^B)^{A\to B}$
$\dfrac{\begin{array}{cc}\mid M & \mid N\\ A \to B & A\end{array}}{B} \to^-$	$(M^{A\to B} N^A)^B$
$\dfrac{\begin{array}{c}\mid M\\ A\end{array}}{\forall_x A} \forall^+ x$ (with var.cond.)	$(\lambda_x M^A)^{\forall_x A}$ (with var.cond.)
$\dfrac{\begin{array}{cc}\mid M & \\ \forall_x A(x) & t\end{array}}{A(t)} \forall^-$	$(M^{\forall_x A(x)} t)^{A(t)}$

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

the forms

$$u\vec{M} \mid \lambda_v M \mid (\lambda_v M) N \vec{L},$$

where u is an assumption variable or assumption constant, v is an assumption variable or object variable, and M, N, L are derivation terms or object terms. Here the final form is not normal: $(\lambda_v M)N\vec{L}$ is called a *β-redex* (for "reducible expression"). It can be reduced by a "conversion". A *conversion* removes a detour in a derivation, i.e., an elimination immediately following an introduction. We consider the following conversions, for derivations written in tree notation and also as derivation terms.

$\to$-conversion.

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

or written as derivation terms

$$(\lambda_u M(u^A)^B)^{A\to B}N^A \quad \beta\text{-converts to} \quad M(N^A)^B.$$

$\forall$-conversion.

$$\dfrac{\dfrac{\begin{array}{c}\mid M\\ A(x)\end{array}}{\forall_x A(x)}\forall^+x \qquad t}{A(t)}\forall^- \quad \beta\text{-converts to} \quad \begin{array}{c}\mid M'\\ A(t)\end{array}$$

or written as derivation terms

$$(\lambda_x M(x)^{A(x)})^{\forall_x A(x)}t \quad \beta\text{-converts to} \quad M(t).$$

The relation $M \succ_1 N$ (*M reduces in one step to N*) means that 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.

Example. Consider assumption variables

$$\begin{array}{lll} x\colon A\to(B\to A)\to A & u\colon A & u'\colon A\\ y\colon A\to B\to A & v\colon B\to A & v'\colon B\\ z\colon A & & \end{array}$$

Then we have derivation terms

$$\begin{array}{rl} S := \lambda_x\lambda_y\lambda_z(xz(yz))\colon & (A\to(B\to A)\to A)\to(A\to B\to A)\to A\to A\\ K := \lambda_u\lambda_v u & :A\to(B\to A)\to A\\ K' := \lambda_{u'}\lambda_{v'}u' & :A\to B\to A \end{array}$$

By the one step reduction relation we obtain

$$\begin{array}{rll} SKK' := & (\lambda_x\lambda_y\lambda_z(xz(yz)))(\lambda_u\lambda_v\, u)(\lambda_{u'}\lambda_{v'}\, u') & \succ_1\\ & (\lambda_y\lambda_z((\lambda_u\lambda_v\, u)z(yz)))(\lambda_{u'}\lambda_{v'}\, u') & \succ_1\\ & (\lambda_y\lambda_z((\lambda_v\, z)(yz)))(\lambda_{u'}\lambda_{v'}\, u') & \succ_1\\ & (\lambda_y\lambda_z\, z)(\lambda_{u'}\lambda_{v'}\, u') & \succ_1 \lambda_z\, z. \end{array}$$

The relation $\succeq$ is the transitive and reflexive closure of $\succ_1$. We list some standard definitions in the setting.

- 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 $M \succeq N$.
- A *reduction sequence* is a sequence $M_0 \succ_1 M_1 \succ_1 M_2 \dots$ (finite or infinite) such that $M_i \succ_1 M_{i+1}$, for all i.

Finite reduction sequences are partially ordered under the initial part relation; the collection of finite reduction sequences starting from a term M forms a tree, the *reduction tree* of M. The branches of this tree may be identified with the collection of all infinite and all terminating finite reduction sequences. A term is *strongly normalizing* if its reduction tree is finite.

One can show that every term is strongly normalizing, and that its normal form is uniquely determined.

1.2.2. The structure of normal derivations. To analyze normal derivations, it will be useful to introduce the notion of a *track* in a proof tree, which makes sense for non-normal derivations as well.

Definition. A *track* of a derivation M is a sequence of formula occurrences (f.o.) $A_0, \dots, A_n$ such that

(a) A_0 is a top f.o. in M;
(b) A_i for $i < n$ is not the minor premise of an instance of $\to^-$, and A_{i+1} is directly below A_i;
(c) A_n is either the minor premise of an instance of $\to^-$, or the conclusion of M.

The *track of order* 0, or *main track*, in a derivation is the (unique) track ending in the conclusion of the whole derivation. 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.

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

Proof. By induction on derivations. □

Now consider a normal derivation M. Since by normality an E-rule cannot have the conclusion of an I-rule as its major premise, the E-rules have to precede the I-rules in a track, so the following is obvious: a track may be divided into an E-part, say $A_0, \dots, A_{i-1}$, a minimal formula A_i, and an I-part $A_{i+1}, \dots, A_n$. In the E-part all rules are E-rules; in the I-part all rules are I-rules; A_i is the conclusion of an E-rule and, if $i < n$, a premise of an I-rule. Tracks are pieces of branches of the tree with successive f.o.'s in the subformula relationship: either A_{i+1} is a subformula of A_i or vice versa. As a result, all formulas in a track $A_0, \dots, A_n$ are subformulas of A_0 or of A_n; and from this, by induction on the order of tracks, we see that

every formula in M is a subformula either of an open assumption or of the conclusion. To summarize:

THEOREM (Subformula property). *In a normal derivation each formula is a subformula of either the end formula or else an assumption formula.*

PROOF. One proves this for tracks of order n, by induction on n. □

REMARK (Long normal form). The minimal formula in a track can be an implication $A \to B$ or a generalization $\forall_x A$. However, we can apply an "η-expansion" and replace the occurrence of $A \to B$ or $\forall_x A$ by

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

Repeating this process we obtain a derivation in "long normal form", all of whose minimal formulas are neither implications nor generalizations.

1.3. 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 the tree models introduced by Beth (1956). Using this concept of a model we will prove soundness and completeness.

1.3.1. 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 $R\vec{t}$ is true in a world k, then $R\vec{t}$ 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 a tree T is a linearly ordered subtree of T with the same root, and a *leaf* of T is a node without successors in T. A tree T is *complete* if every node in T has an immediate successor, i.e., T has no leaves.

For the proof of the completeness theorem, the full tree over $\{0, 1\}$ (whose branches constitute Cantor space) 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 $\mathcal{L}$.

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 non-empty 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. There is no special requirement set on $I_1(\bot, k)$. (Recall that minimal logic places no particular constraints on falsum $\bot$.) We write $R^{\mathcal{T}}(\vec{a}, k)$ for $\vec{a} \in I_1(R, k)$, and $|\mathcal{T}|$ to denote the domain D.

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

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 $[x_1 := a_1, \ldots, x_n := a_n]$ or else as $[a_1/x_1, \ldots, a_n/x_n]$, with distinct $x_1, \ldots, x_n$. If η is an assignment in D and $a \in D$, let η^a_x be the assignment in D mapping x to a and coinciding with η elsewhere:

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

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 whose variables lie in $\mathrm{dom}(\eta)$ by

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

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 $\forall_{k' \succeq k}(\mathrm{lh}(k') = \mathrm{lh}(k) + n \to A)$.

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

Thus in the atomic, disjunctive and existential cases, the set of k' whose length is $\mathrm{lh}(k) + n$ acts as a "bar" in the complete tree. Note that the implicational case is treated differently, and refers to the "unbounded future".

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.

1.3.2. 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 true as well:

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 $R\vec{t}$. Assume

$$\forall_{k' \succeq_n k}(k' \Vdash R\vec{t}\,),$$

hence by definition

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

Since T is a finitely branching tree,

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

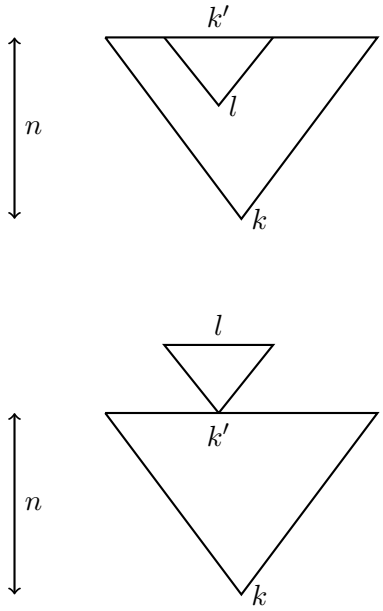
Hence $k \Vdash R\vec{t}$.

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

Case $A \to B$. Assume that for all $k' \succeq k$ with $\mathrm{lh}(k') = \mathrm{lh}(k) + n$ we have $k' \Vdash A \to B$. The goal is $k \Vdash A \to B$, i.e.,

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

Let $l \succeq k$ and $l \Vdash A$. We must show $l \Vdash B$. *Case* $\mathrm{lh}(l) \leq \mathrm{lh}(k')$:
Then $k' \Vdash B$ for all these k', hence $l \Vdash B$ by IH for B. *Case* $\mathrm{lh}(k') < \mathrm{lh}(l)$:



Because of $k' \Vdash A \to B$ and $l \Vdash A$ we obtain $l \Vdash B$, by definition of $\Vdash$.

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

1.3.3. 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, $t, r(x)$ terms, $A(x)$ 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 any node and η an assignment in $|\mathcal{T}|$, it follows that $\mathcal{T}, k \Vdash \Gamma[\eta]$ implies $\mathcal{T}, k \Vdash A[\eta]$.*

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 $\vee_0^+ \colon A \to A \vee B$. 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 $\vee_1^+ \colon B \to A \vee B$ is symmetric.

Case $\vee^- \colon A \vee B \to (A \to C) \to (B \to C) \to C$. 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 $\exists^+ \colon A \to \exists_x A$. 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 $\exists^- \colon \exists_x A \to \forall_x(A \to B) \to B$ 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 induction hypothesis.

Case $\to^-$. Assume $k \Vdash \Gamma$. The induction hypothesis 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

$k \Vdash \Gamma[\eta_x^a]$ by the coincidence lemma, since $x \notin \mathrm{FV}(\Gamma)$,

$k \Vdash A[\eta_x^a]$ by induction hypothesis.

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

$k \Vdash (\forall_x A(x))[\eta]$ by induction hypothesis,

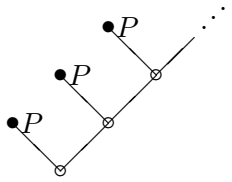
$k \Vdash A(x)[\eta_x^{\eta(t)}]$ by definition,

$k \Vdash A(t)[\eta]$ by the substitution lemma.

This concludes the proof. □

1.3.4. Counter models. With soundness at hand, it is easy to build counter models proving that certain formulas are underivable in minimal or intuitionistic logic. A *tree model for intuitionistic logic* is a tree model $\mathcal{T} = (D, I_0, I_1)$ in which $I_1(\bot, k)$ is false for all k.

As an example we show that $\not\vdash_i \neg\neg P \to P$. Assume $\vdash_i \neg\neg P \to P$, i.e., Efq $\vdash \neg\neg P \to P$. We will obtain a contradiction from this assumption. For simplicity we assume that from Efq we have only used $\bot \to P$ and say $\bot \to Q$ for some Q. We can now substitute $\bot$ for Q everywhere and obtain a derivation in minimal logic, since $\bot \to \bot$ is immediately derivable. Hence we have $\vdash (\bot \to P) \to \neg\neg P \to P$. We can now obtain the desired contradiction using a tree model determined by the figure below. Next to every node we write all propositions forced at that node.



This is a tree model because monotonicity clearly holds. Observe also that $I_1(\bot, k)$ is false at all nodes k. Hence this is an intuitionistic tree model. By the definition of forcing we have

(i) $\bot \to P$ is forced at every node.
(ii) $P \to \bot$ (i.e., $\neg P$) is never forced.
(iii) $\neg\neg P$ is forced at every node.
(iv) The root node does not force P, since there are arbitrarily long $\circ$-nodes.

This is the desired contradiction to the Soundness Theorem.

The model also shows that the *Peirce formula* $((P \to Q) \to P) \to P$ is not derivable in intuitionistic logic.

1.3.5. 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 already gives "(a) implies (b)". 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) whose domain D is the set of all terms of the underlying language, with the property that $\Gamma \vdash B$ is

equivalent to $\mathcal{T}, \langle\rangle \Vdash B[\mathrm{id}]$. $\mathcal{T}$ will depend on Γ. We can assume here that Γ and A are closed.

In order to define $\mathcal{T}$, we will need an enumeration $A_0, A_1, A_2, \ldots$ of the underlying language $\mathcal{L}$ (assumed countable), in which every formula occurs infinitely often. We also fix an enumeration $x_0, x_1, \ldots$ of distinct variables. Since Γ is countable it can we written $\Gamma = \bigcup_n \Gamma_n$ 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 $\leq n$ 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 the fact that 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

$$\forall_{k' \succeq_n k} (\Gamma, \Delta_{k'} \vdash B) \to \Gamma, \Delta_k \vdash B, \quad \text{provided } \mathrm{FV}(B) \subseteq V_k. \tag{13}$$

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 schema $\vee^-$. 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

$$\Gamma, \Delta_k \vdash B \to \exists_n \forall_{k' \succeq_n k} (B \in \Delta_{k'}), \quad \text{provided } \mathrm{FV}(B) \subseteq V_k. \tag{14}$$

Choose $n \geq \mathrm{lh}(k)$ 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, $I_0(f)(\vec{t}\,) := f\vec{t}$ and

$$R^{\mathcal{T}}(\vec{t}, k) = I_1(R, k)(\vec{t}\,) := (R\vec{t} \in \Delta_k).$$

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

Now write $k \Vdash B$ for $\mathcal{T}, k \Vdash B[\mathrm{id}]$. We show:

Claim. $\Gamma, \Delta_k \vdash B \leftrightarrow k \Vdash B$ *provided* $\mathrm{FV}(B) \subseteq V_k$.

The proof is by induction on B.

Case $R\vec{t}$. Assume $\mathrm{FV}(R\vec{t}\,) \subseteq V_k$. The following are equivalent:

$\Gamma, \Delta_k \vdash R\vec{t}$,

$\exists_n \forall_{k' \succeq_n k} (R\vec{t} \in \Delta_{k'})$ by (14) and (13),

$\exists_n \forall_{k' \succeq_n k} R^{\mathcal{T}}(\vec{t}, k')$ by definition of $\mathcal{T}$,

$k \Vdash R\vec{t}$ by definition of $\Vdash$, since $t^{\mathcal{T}}[\mathrm{id}] = t$.

Case $B \vee C$. Assume $\mathrm{FV}(B \vee C) \subseteq V_k$. For the implication $\to$ 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 induction hypothesis

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

Then by definition we have $k \Vdash B \vee C$. For the reverse implication $\leftarrow$ argue as follows.

$k \Vdash B \vee C$,

$\exists_n \forall_{k' \succeq_n k} (k' \Vdash B \vee k' \Vdash C)$,

$\exists_n \forall_{k' \succeq_n k} ((\Gamma, \Delta_{k'} \vdash B) \vee (\Gamma, \Delta_{k'} \vdash C))$ by induction hypothesis,

$\exists_n \forall_{k' \succeq_n k} (\Gamma, \Delta_{k'} \vdash B \vee C)$,

$\Gamma, \Delta_k \vdash B \vee C$ by (13).

Case $B \wedge C$. This is evident.

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

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

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

For $\leftarrow$ let $k \Vdash B \to C$, i.e., $\forall_{k' \succeq k}(k' \Vdash B \to k' \Vdash C)$. We show that $\Gamma, \Delta_k \vdash B \to C$, using (13). 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 $\Gamma, \Delta_{k'} \vdash B \to C$.

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

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

Case $\forall_x B(x)$. Assume $\mathrm{FV}(\forall_x B(x)) \subseteq V_k$. For $\to$ let $\Gamma, \Delta_k \vdash \forall_x B(x)$. Fix a term t. Then $\Gamma, \Delta_k \vdash B(t)$. Choose $n \geq \mathrm{lh}(k)$ such that $\mathrm{FV}(B(t)) \subseteq V_{k'}$ for all k' with $\mathrm{lh}(k') = n$. Then $\forall_{k' \succeq_m k}(\Gamma, \Delta_{k'} \vdash B(t))$ with $m := n - \mathrm{lh}(k)$, hence $\forall_{k' \succeq_m k}(k' \Vdash B(t))$ by induction hypothesis, hence $k \Vdash B(t)$ by the covering lemma. This holds for every term t, hence $k \Vdash \forall_x B(x)$.

For $\leftarrow$ 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 induction hypothesis, 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 (13).

Case $\exists_x B(x)$. Assume $\mathrm{FV}(\exists_x B(x)) \subseteq V_k$. For $\to$ 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 induction hypothesis 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 $k \Vdash \exists_x B(x)$.

For $\leftarrow$ assume $k \Vdash \exists_x B(x)$. Then $\forall_{k' \succeq_n k} \exists_{t \in \mathrm{Ter}}(k' \Vdash B(x)[\mathrm{id}^t_x])$ 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 induction hypothesis for $B(t)$ yields

$$\begin{aligned}
&\forall_{k''\succeq_{m_0+n}k}\exists_{t\in\mathrm{Ter}}\,(\Gamma,\Delta_{k''}\vdash B(t)),\\
&\forall_{k''\succeq_{m_0+n}k}\,(\Gamma,\Delta_{k''}\vdash\exists_x B(x)),\\
&\Gamma,\Delta_k\vdash\exists_x B(x) \qquad\qquad \text{by (13)},
\end{aligned}$$

and this completes the proof of the claim.

Now we can finish the proof of the completeness theorem by showing that (b) implies (a). 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 the claim (since each formula in Γ is derivable from Γ). Hence $\mathcal{T},\langle\rangle\Vdash A[\mathrm{id}]$ by (b) and therefore $\Gamma\vdash A$ by the claim again. □