Chapter 6

Gentzen's Cut Elimination Theorem And Applications

6.1 Introduction

The rules of the Gentzen system G (given in definition 5.4.1) were chosen mainly to facilitate the design of the *search* procedure. The Gentzen system LK' given in Section 3.6 can be extended to a system for first-order logic called LK, which is more convenient for constructing proofs in working downward, and is also useful for proof-theoretical investigations. In particular, the system LK will be used to prove three classical results, Craig's theorem, Beth's theorem and Robinson's theorem, and will be used in Chapter 7 to derive a constructive version of Herbrand's theorem.

The main result about LK is Gentzen's cut elimination theorem. An entirely proof-theoretic argument (involving only algorithmic proof transformation steps) of the cut elimination can be given, but it is technically rather involved. Rather than giving such a proof (which can be found in Szabo, 1969, or Kleene, 1952), we will adopt the following compromise: We give a rather simple semantic proof of Gentzen's cut elimination theorem (using the completeness theorem) for LK, and we give a constructive proof of the cut elimination for a Gentzen system $G1^{nnf}$ simpler than LK (by constructive, we mean that an algorithm for converting a proof into a cut-free proof is actually given). The sequents of the system $G1^{nnf}$ are pairs of sets of formulae in negation normal form. This system is inspired from Schwichtenberg (see Barwise, 1977). The cut elimination theorem for the system $G1^{nnf}_{=}$ which includes axioms for equality is also given.

Three applications of the cut elimination theorem will also be given:

Craig's interpolation theorem,

Beth's definability theorem and

Robinson's joint consistency theorem.

These are classical results of first-order logic, and the proofs based on cut elimination are constructive and elegant. Beth's definability theorem also illustrates the subtle interaction between syntax and semantics.

A new important theme emerges in this chapter, and will be further elaborated in Chapter 7. This is the notion of *normal form* for proofs. Gentzen's cut elimination theorem (for LK or LK_e) shows that for every provable sequent $\Gamma \to \Delta$, there is a proof in normal form, in the sense that in the system LK, the proof does not have any cuts, and for the system LK_e, it has only atomic cuts. In the next chapter, it will be shown that this normal form can be improved if the formulae in the sequent are of a certain type (prenex form or negation normal form). This normal form guaranteed by Gentzen's Sharpened Hauptsatz is of fundamental importance, since it reduces provabilily in first-order logic to provability in propositional logic.

Related to the concept of normal form is the concept of *proof transformation*. Indeed, a constructive way of obtaining proofs in normal form is to perform a sequence of proof transformations.

The concepts of normal form for proofs and of proof transformation are essential. In fact, it turns out that the completeness results obtained in the remaining chapters will be obtained via proof transformations.

First, we consider the case of a first-order language without equality. In this Chapter, it is assumed that no variable occurs both free and bound in any sequent (or formula).

6.2 Gentzen System LK for Languages Without Equality

The system LK is obtained from LK' by adding rules for quantified formulae.

6.2.1 Syntax of LK

The inference rules and axioms of LK are defined as follows.

Definition 6.2.1 Gentzen system LK. The system LK consists of *structural rules*, the *cut rule*, and of *logical rules*. The letters Γ, Δ, Λ, Θ stand for arbitrary (possibly empty) sequences of formulae and A, B for arbitrary formulae.

(1) Structural rules:

(i) Weakening:

$$\frac{\Gamma \to \Delta}{A, \Gamma \to \Delta}\ (left) \qquad \frac{\Gamma \to \Delta}{\Gamma \to \Delta, A}\ (right)$$

A is called the *weakening formula*

(ii) Contraction:

$$\frac{A, A, \Gamma \to \Delta}{A, \Gamma \to \Delta}\ (left) \qquad \frac{\Gamma \to \Delta, A, A}{\Gamma \to \Delta, A}\ (right)$$

(iii) Exchange:

$$\frac{\Gamma, A, B, \Delta \to \Lambda}{\Gamma, B, A, \Delta \to \Lambda}\ (left) \qquad \frac{\Gamma \to \Delta, A, B, \Lambda}{\Gamma \to \Delta, B, A, \Lambda}\ (right)$$

(2) Cut rule:

$$\frac{\Gamma \to \Delta, A \qquad A, \Lambda \to \Theta}{\Gamma, \Lambda \to \Delta, \Theta}$$

A is called the *cut formula* of this inference.

(3) Logical rules:

$$\frac{A, \Gamma \to \Delta}{A \wedge B, \Gamma \to \Delta}\ (\wedge : left) \quad \text{and} \quad \frac{B, \Gamma \to \Delta}{A \wedge B, \Gamma \to \Delta}\ (\wedge : left)$$

$$\frac{\Gamma \to \Delta, A \qquad \Gamma \to \Delta, B}{\Gamma \to \Delta, A \wedge B}\ (\wedge : right)$$

$$\frac{A, \Gamma \to \Delta \qquad B, \Gamma \to \Delta}{A \vee B, \Gamma \to \Delta}\ (\vee : left)$$

$$\frac{\Gamma \to \Delta, A}{\Gamma \to \Delta, A \vee B}\ (\vee : right) \quad \text{and} \quad \frac{\Gamma \to \Delta, B}{\Gamma \to \Delta, A \vee B}\ (\vee : right)$$

$$\frac{\Gamma \to \Delta, A \qquad B, \Lambda \to \Theta}{A \supset B, \Gamma, \Lambda \to \Delta, \Theta}\ (\supset : left) \qquad \frac{A, \Gamma \to \Delta, B}{\Gamma \to \Delta, A \supset B}\ (\supset : right)$$

$$\frac{\Gamma \rightarrow \Delta, A}{\neg A, \Gamma \rightarrow \Delta}\ (\neg : left) \qquad \frac{A, \Gamma \rightarrow \Delta}{\Gamma \rightarrow \Delta, \neg A}\ (\neg : right)$$

In the rules above, $A \vee B$, $A \wedge B$, $A \supset B$, and $\neg A$ are called the *principal formulae* and A, B the *side formulae* of the inference.

In the quantifier rules below, x is any variable and y is any variable free for x in A and not free in A, unless $y = x$ ($y \notin FV(A) - \{x\}$). The term t is any term free for x in A.

$$\frac{A[t/x], \Gamma \rightarrow \Delta}{\forall x A, \Gamma \rightarrow \Delta}\ (\forall : left) \qquad \frac{\Gamma \rightarrow \Delta, A[y/x]}{\Gamma \rightarrow \Delta, \forall x A}\ (\forall : right)$$

$$\frac{A[y/x], \Gamma \rightarrow \Delta}{\exists x A, \Gamma \rightarrow \Delta}\ (\exists : left) \qquad \frac{\Gamma \rightarrow \Delta, A[t/x]}{\Gamma \rightarrow \Delta, \exists x A}\ (\exists : right)$$

Note that in both the $(\forall : right)$-rule and the $(\exists : left)$-rule, the variable y does *not* occur free in the lower sequent. In these rules, the variable y is called the *eigenvariable* of the inference. The condition that the eigenvariable does not occur free in the conclusion of the rule is called the *eigenvariable condition*. The formula $\forall x A$ (or $\exists x A$) is called the *principal formula* of the inference, and the formula $A[t/x]$ (or $A[y/x]$) the *side formula* of the inference.

The *axioms* of the system LK are all sequents of the form $A \rightarrow A$.

Note that since the system LK contains the exchange rules, the order of the formulae in a sequent is really irrelevant. Hence, we can view a sequent as a pair of multisets (as defined in problem 2.1.8).

Proof trees and *deduction tree* are defined inductively as in definition 3.4.5, but with the rules of the system LK given in definition 6.2.1. If a sequent has a proof in the system G we say that it is *G-provable* and similarly, if it is provable in the system LK, we say that it is *LK-provable*. The system obtained from LK by removing the cut rule is denoted by $LK - \{cut\}$. We also say that a sequent is LK-provable without a cut if it has a proof tree using the rules of the system $LK - \{cut\}$.

6.2.2 The Logical Equivalence of the Systems G, LK, and $LK - \{cut\}$

We now show that the systems G and LK are logically equivalent. We will in fact prove a stronger result, namely that G, $LK - \{cut\}$ and LK are equivalent. First, we show that the system LK is sound.

Lemma 6.2.1 Every axiom of LK is valid. For every rule of LK, if the premises of the rule are valid, then the conclusion of the rule is valid. Every LK-provable sequent is valid.

Proof: The proof uses the induction principle for proofs and is straightforward. □

lemma 6.2.1 differs from lemma 5.4.3 in the following point: It is not necessarily true that if the conclusion of a rule is valid, then the premises of that rule are valid.

Theorem 6.2.1 (Logical equivalence of G, LK, and $LK - \{cut\}$) There is an algorithm to convert any LK-proof of a sequent $\Gamma \rightarrow \Delta$ into a G-proof. There is an algorithm to convert any G-proof of a sequent $\Gamma \rightarrow \Delta$ into a proof using the rules of $LK - \{cut\}$.

Proof: If $\Gamma \rightarrow \Delta$ has an LK-proof, by lemma 6.2.1, $\Gamma \rightarrow \Delta$ is valid. By theorem 5.5.1, $\Gamma \rightarrow \Delta$ has a G-proof given by the procedure *search*. Conversely, using the induction principle for G-proofs we show that every G-proof can be converted to an $(LK - \{cut\})$-proof. The proof is similar to that of theorem 3.6.1. Every G-axiom $\Gamma \rightarrow \Delta$ contains some common formula A, and by application of the weakening and the exchange rules, an $(LK - \{cut\})$-proof of $\Gamma \rightarrow \Delta$ can be obtained from the axiom $A \rightarrow A$. Next, we have to show that every application of a G-rule can be replaced by a sequence of $(LK - \{cut\})$-rules. There are twelve cases to consider. Note that the G-rules $\land : right$, $\lor : left$, $\supset: right$, $\supset: left$, $\neg : right$, $\neg : left$, $\forall : right$ and $\exists : left$ can easily be simulated in $LK - \{cut\}$ using the exchange, contraction, and corresponding $(LK - \{cut\})$-rules. The rules $\lor : right$ and $\land : left$ are handled as in theorem 3.6.1. Finally, we show how the G-rule $\forall : left$ can be transformed into a sequence of $(LK - \{cut\})$-rules, leaving the $\exists : right$ case as an exercise.

$$
\begin{array}{cl}
\Gamma, A[t/x], \forall xA, \Delta \rightarrow \Lambda & \\
\hline
\text{(several exchanges)} & \\
\hline
A[t/x], \forall xA, \Gamma, \Delta \rightarrow \Lambda & \\
\hline
\forall xA, \forall xA, \Gamma, \Delta \rightarrow \Lambda & (\forall : left) \\
\hline
\forall xA, \Gamma, \Delta \rightarrow \Lambda & (contraction) \\
\hline
\text{(several exchanges)} & \\
\hline
\Gamma, \forall xA, \Delta \rightarrow \Lambda &
\end{array}
$$

The above $(LK - \{cut\})$-derivation simulates the G-rule $\forall : left$. This conclude the proof of the theorem. □

Corollary (Gentzen Hauptsatz for LK) A sequent is LK-provable if and only if it is LK-provable without a cut.

Note that the *search* procedure together with the method indicated in theorem 6.2.1 actually provides an algorithm to construct a cut-free LK-proof from an LK-proof with cuts. Gentzen proved the above result by a very different method in which an LK-proof is (recursively) transformed into an LK-proof without cut. Gentzen's proof is entirely constructive since it does not use any semantic arguments and can be found either in Takeuti, 1975, Kleene, 1952, or in Gentzen's original paper in Szabo, 1969.

In the next section, we discuss the case of first-order languages with equality symbol.

PROBLEMS

6.2.1. Give LK-proofs for the following formulae:

$$\forall xA \supset A[t/x],$$
$$A[t/x] \supset \exists xA,$$
where t is free for x in A.

6.2.2. Let x, y be any distinct variables. Let A, B be any formulae, C, D any formulae not containing the variable x free, and let E be any formula such that x is free for y in E. Give proof trees for the following formulae:

$$\begin{array}{cc}
\forall xC \equiv C & \exists xC \equiv C \\
\forall x\forall yA \equiv \forall y\forall xA & \exists x\exists yA \equiv \exists y\exists xA \\
\forall x\forall yE \supset \forall xE[x/y] & \exists xE[x/y] \supset \exists x\exists yE \\
\forall xA \supset \exists xA & \\
\exists x\forall yA \supset \forall y\exists xA &
\end{array}$$

6.2.3. First, give G-proofs for the formulae of problem 6.2.2, and then convert them into LK-proofs using the method of theorem 6.2.1.

6.2.4. Give an LK-proof for

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

Give an LK-proof for

$$(B \supset (C \supset A)), (A \supset (B \supset C)) \rightarrow (B \supset (A \equiv C))$$

using the cut rule, and another LK-proof without cut.

6.2.5. Given a set S of formulae, let $Des(S)$ be the set of immediate descendants of formulae in S as defined in problem 5.5.18, and define S^n by induction as follows:

$$S^0 = S;$$
$$S^{n+1} = Des(S^n).$$
$$\text{Let } S^* = \bigcup_{n\geq 0} S^n.$$

S^* is called the set of *descendants* of S.

(a) Every sequent $\Gamma \to \Delta$ corresponds to the set of formulae $\Gamma \cup \{\neg B \mid B \in \Delta\}$. Prove that for every deduction tree for a sequent $\Gamma_0 \to \Delta_0$, the union of the sets of formulae $\Gamma \cup \{\neg B \mid B \in \Delta\}$ for all sequents $\Gamma \to \Delta$ occurring in that tree is a subset of S^*, where $S = \Gamma_0 \cup \{\neg B \mid B \in \Delta_0\}$ (this is called the *subformula property*).

(b) Deduce from (a) that not all formulae are provable.

6.3 The Gentzen System LK_e With Equality

We now generalize the system LK to include equality.

6.3.1 Syntax of LK_e

The rules and axioms of LK_e are defined as follows.

Definition 6.3.1 (The Gentzen system LK_e) The Gentzen system LK_e is obtained from the system LK by adding the following sequents known as *equality axioms*:

Let $t, s_1, ..., s_n, t_1, ..., t_n$ be arbitrary **L**-terms. For every term t, the sequent

$$\to t \doteq t$$

is an axiom.

For every n-ary function symbol f,

$$s_1 \doteq t_1, s_2 \doteq t_2, ..., s_n \doteq t_n \to fs_1...s_n \doteq ft_1...t_n$$

is an axiom.

For every n-ary predicate symbol P (including $\doteq$),

$$s_1 \doteq t_1, s_2 \doteq t_2, ..., s_n \doteq t_n, Ps_1...s_n \to Pt_1...t_n$$

is an axiom.

It is easily shown that these axioms are valid. In order to generalize theorem 6.2.1 it is necessary to define the concept of an atomic cut.

Definition 6.3.2 If the cut formula of a cut in an LK_e-proof is an atomic formula, the cut is called *atomic*. Otherwise, it is called an *essential cut*.

theorem 6.2.1 can now be generalized, provided that we allow atomic cuts. However, some technical lemmas including an exchange lemma will be needed in the proof.

6.3.2 A Permutation Lemma for the System $G_=$

The following lemma holds in $G_=$.

Lemma 6.3.1 Given a $G_=$-proof T, there is a $G_=$-proof T' such that all leaf sequents of T' contain either atomic formulae or quantified formulae (that is, formulae of the form $\forall xB$ or $\exists xB$).

Proof: Given a nonatomic formula A not of the form $\forall xB$ nor $\exists xB$, define its *weight* as the number of logical connectives $\wedge$, $\vee$, $\neg$, $\supset$ in it. The weight of an atomic formula and of a quantified formula is 0. The weight of a sequent is the sum of the weights of the formulae in it. The weight of a proof tree is the maximum of the weights of its leaf sequents. We prove the lemma by induction on the weight of the proof tree T. If $weight(T) = 0$, the lemma holds trivially. Otherwise, we show how the weight of each leaf sequent of T whose weight is nonzero can be decreased. This part of the proof proceeds by cases. We cover some of the cases, leaving the others as an exercise.

Let S be any leaf sequent of T whose weight is nonzero. Then, for some formula A, S is of the form

$$\Gamma_1, A, \Gamma_2 \to \Delta_1, A, \Delta_2.$$

If A is the only formula in S whose weight is nonzero, we can extend the leaf S as follows:

Case 1: A is of the form $B \wedge C$.

$$\cfrac{\cfrac{\Gamma_1, B, C, \Gamma_2 \to \Delta_1, B, \Delta_2 \qquad \Gamma_1, B, C, \Gamma_2 \to \Delta_1, C, \Delta_2}{\Gamma_1, B, C, \Gamma_2 \to \Delta_1, B \wedge C, \Delta_2}}{\Gamma_1, B \wedge C, \Gamma_2 \to \Delta_1, B \wedge C, \Delta_2}$$

The weights of the new leaf sequents are strictly smaller than the weight of S.

Case 2: A is of the form $\neg B$.

$$\cfrac{\cfrac{B, \Gamma_1, \Gamma_2 \to B, \Delta_1, \Delta_2}{\Gamma_1, \Gamma_2 \to B, \Delta_1, \neg B, \Delta_2}}{\Gamma_1, \neg B, \Gamma_2 \to \Delta_1, \neg B, \Delta_2}$$

The weight of the new leaf sequent is strictly smaller than the weight of S.

Case 3: A is of the form $B \supset C$.

$$\cfrac{\cfrac{B, \Gamma_1, \Gamma_2 \to B, C, \Delta_1, \Delta_2 \qquad C, B, \Gamma_1, \Gamma_2 \to C, \Delta_1, \Delta_2}{B, \Gamma_1, B \supset C, \Gamma_2 \to C, \Delta_1, \Delta_2}}{\Gamma_1, B \supset C, \Gamma_2 \to \Delta_1, B \supset C, \Delta_2}$$

The weights of the new leaf sequents are strictly smaller than the weight of S.

We leave the case in which A is of the form $(C \vee D)$ as an exercise.

If A is not the only formula in $\Gamma_1, A, \Gamma_2 \to \Delta_1, A, \Delta_2$ whose weight is nonzero, if we apply the corresponding rule to that formula, we obtain a sequent S_1 or two sequents S_1 and S_2 still containing A on both sides of the arrow, and whose weight is strictly smaller than the weight of S.

Now, if we apply the above transformations to all leaf sequents of T whose weight is nonzero, since the weight of each new leaf sequent is strictly smaller than the weight of some leaf sequent of T, we obtain a tree T' whose weight is strictly smaller than the weight of T. We conclude by applying the induction hypothesis to T'. This completes the proof. $\square$

Lemma 6.3.2 (i) Given a $G_=$-proof tree T for a sequent $\Gamma \to A \wedge B, \Delta$ satisfying the conclusion of lemma 6.3.1, another $G_=$-proof T' can be constructed such that, $depth(T) = depth(T')$, and the rule applied at the root of T' is the $\wedge : right$ rule applied to the occurrence of $A \wedge B$ to the right of $\to$.

(ii) Given a $G_=-$proof tree T of a sequent $\Gamma, A \supset B \to \Delta$ satisfying the conclusion of lemma 6.3.1, another $G_=$-proof tree T' can be constructed such that, $depth(T) = depth(T')$, and the rule applied at the root of T' is the $\supset$: $left$ rule applied to the occurrence of $A \supset B$ to the left of $\to$.

Proof: (i) Let S be the initial subtree of T obtained by deleting the descendants of every node closest to the root of T, where the $\wedge : right$ rule is applied to the formula $A \wedge B$ to the right of $\to$ in the sequent labeling that node. Since the proof tree T satisfies the condition of lemma 6.3.1, the rule $\wedge : right$ is applied to each occurrence of $A \wedge B$ on the right of $\to$. Hence, the tree T has the following shape:

Tree T

$$
\cfrac{\cfrac{S_1}{\Gamma_1 \rightarrow A, \Delta_1} \qquad \cfrac{T_1}{\Gamma_1 \rightarrow B, \Delta_1}}{\Gamma_1 \rightarrow A \wedge B, \Delta_1} \qquad \cfrac{\cfrac{S_m}{\Gamma_m \rightarrow A, \Delta_m} \qquad \cfrac{T_m}{\Gamma_m \rightarrow B, \Delta_m}}{\Gamma_m \rightarrow A \wedge B, \Delta_m}
$$

$$S$$

$$\Gamma \rightarrow A \wedge B, \Delta$$

where the tree with leaves $\Gamma_1 \rightarrow A \wedge B, \Delta_1$,..., $\Gamma_m \rightarrow A \wedge B, \Delta_m$ is the tree S, and the subtrees S_1, T_1,..., S_m, T_m are proof trees. Let S' be the tree obtained from S by replacing every occurrence in S of the formula $A \wedge B$ to the right of $\rightarrow$ by A, and S'' the tree obtained from S by replacing every occurrence of $A \wedge B$ on the right of $\rightarrow$ by B. Since no rule is applied to an occurrence of $A \wedge B$ on the right of $\rightarrow$ in S, both S' and S'' are well defined. The following proof tree T' satisfies the conditions of the lemma:

Tree T'

$$
\cfrac{S_1}{\Gamma_1 \rightarrow A, \Delta_1} \qquad \cfrac{S_m}{\Gamma_m \rightarrow A, \Delta_m} \qquad \cfrac{T_1}{\Gamma_1 \rightarrow B, \Delta_1} \qquad \cfrac{T_m}{\Gamma_m \rightarrow B, \Delta_m}
$$

$$S' \qquad\qquad\qquad S''$$

$$
\cfrac{\Gamma \rightarrow A, \Delta \qquad \Gamma \rightarrow B, \Delta}{\Gamma \rightarrow A \wedge B, \Delta}
$$

It is clear that $depth(T) = depth(T')$.

(ii) The proof is similar to that of (i). The proof tree T can be converted to T' as shown:

Tree T

$$
\cfrac{\cfrac{S_1}{\Gamma_1 \rightarrow A, \Delta_1} \qquad \cfrac{T_1}{B, \Gamma_1 \rightarrow \Delta_1}}{\Gamma_1, A \supset B \rightarrow \Delta_1} \qquad \cfrac{\cfrac{S_m}{\Gamma_m \rightarrow A, \Delta_m} \qquad \cfrac{T_m}{B, \Gamma_m \rightarrow \Delta_m}}{\Gamma_m, A \supset B \rightarrow \Delta_m}
$$

$$S$$

$$\Gamma, A \supset B \rightarrow \Delta$$

Tree T'

$$\frac{S_1}{\Gamma_1 \to A, \Delta_1} \qquad \frac{S_m}{\Gamma_m \to A, \Delta_m} \qquad \frac{T_1}{B, \Gamma_1 \to \Delta_1} \qquad \frac{T_m}{B, \Gamma_m \to \Delta_m}$$

$$S' \qquad\qquad S''$$

$$\frac{\Gamma \to A, \Delta \qquad B, \Gamma \to \Delta}{\Gamma, A \supset B \to \Delta}$$

Note that $depth(T) = depth(T')$. $\square$

Lemma 6.3.3 Every $G_=$-proof tree T can be converted to a proof tree T' of the same sequent such that the rule applied to every sequent of the form $\Gamma \to A \wedge B, \Delta$ or $\Gamma, C \supset D \to \Delta$ is either the $\wedge : right$ rule applied to the occurrence of $A \wedge B$ to the right of $\to$, or the $\supset$: $left$ rule applied to the occurrence of $C \supset D$ to the left of $\to$. Furthermore, if T satisfies the conditions of lemma 6.3.1, then T' has the same depth as T.

Proof: First, using lemma 6.3.1, we can assume that T has been converted to a proof tree such that in all leaf sequents, all formulae are either atomic or quantified formulae. Then, since lemma 6.3.2 preserves the depth of such proof trees, we conclude by induction on the depth of proof trees using lemma 6.3.2. Since the transformations of lemma 6.3.2 are depth preserving, the last clause of the lemma follows. $\square$

6.3.3 Logical equivalence of $G_=$, LK_e, and LK_e Without Essential Cuts: Gentzen's Hauptsatz for LK_e Without Essential Cuts

The generalization of theorem 6.2.1 is the following.

Theorem 6.3.1 (Logical equivalence of $G_=$, LK_e, and LK_e without essential cuts) There is an algorithm to convert any LK_e-proof of a sequent $\Gamma \to \Delta$ into a $G_=$-proof. There is an algorithm to convert any $G_=$-proof of a sequent $\Gamma \to \Delta$ into an LK_e-proof without essential cuts.

Proof: The proof is similar to that of theorem 6.2.1. The proof differs because atomic cuts cannot be eliminated. By lemma 6.3.1, we can assume that the axioms of proof trees are either atomic formulae or quantified formulae. We proceed by induction on $G_=$-proof trees having axioms of this form. The base case is unchanged and, for the induction step, only the equality rules of definition 6.2.1 need to be considered, since the other rules are handled as in theorem 6.2.1.

(i) The rule

$$\frac{\Gamma, t \doteq t \to \Delta}{\Gamma \to \Delta}$$

is simulated in LK_e using an atomic cut as follows:

$$\dfrac{\to t \doteq t \qquad \dfrac{\dfrac{\Gamma, t \doteq t \to \Delta}{\text{exchanges}}}{t \doteq t, \Gamma \to \Delta}}{\Gamma \to \Delta} \quad \text{atomic cut}$$

(ii) The root of the $G_=$-proof tree T is the conclusion of the rule

$$\frac{\Gamma, (s_1 \doteq t_1) \wedge ... \wedge (s_n \doteq t_n) \supset (fs_1...s_n \doteq ft_1...t_n) \to \Delta}{\Gamma \to \Delta}$$

If the premise of this rule is an axiom in $G_=$, this sequent contains a same formula A on both sides, and an LK_e-proof without cuts can be obtained from $A \to A$ using the exchange and weakening rules. Otherwise, using lemma 6.3.3, an LK_e-derivation with only atomic cuts can be constructed as follows. By lemma 6.3.3, the proof tree T is equivalent to a proof tree of same depth having the following shape:

$$\dfrac{\dfrac{T_{n-1}}{\Gamma \to s_{n-1} \doteq t_{n-1}, \Delta} \qquad \dfrac{T_n}{\Gamma \to s_n \doteq t_n, \Delta}}{\Gamma \to s_{n-1} \doteq t_{n-1} \wedge s_n \doteq t_n, \Delta}$$

$$...$$

$$\dfrac{\dfrac{\dfrac{T_1}{\Gamma \to s_1 \doteq t_1, \Delta} \Gamma \to s_2 \doteq t_2 \wedge ... \wedge s_n \doteq t_n, \Delta}{\Gamma \to s_1 \doteq t_1 \wedge ... \wedge s_n \doteq t_n, \Delta} \qquad \dfrac{T_0}{fs_1...s_n \doteq ft_1...t_n, \Gamma \to \Delta}}{\Gamma, s_1 \doteq t_1 \wedge ... \wedge s_n \doteq t_n \supset fs_1...s_n \doteq ft_1...t_n \to \Delta}$$

Using the axiom

$$s_1 \doteq t_1, ..., s_n \doteq t_n \to fs_1...s_n \doteq ft_1...t_n$$

and applying the induction hypothesis to the $G_=$-trees T_0, T_1,..., T_n, an LK_e-proof without essential cuts can be constructed:

$$\frac{\dfrac{T'_n}{\Gamma \to \Delta, s_n \doteq t_n} \qquad \dfrac{s_1 \doteq t_1, ..., s_n \doteq t_n \to fs_1...s_n \doteq ft_1...t_n}{s_n \doteq t_n, ..., s_1 \doteq t_1 \to fs_1...s_n \doteq ft_1...t_n}}{s_{n-1} \doteq t_{n-1}, ..., s_1 \doteq t_1, \Gamma \to fs_1...s_n \doteq ft_1...t_n, \Delta}$$

...

$$\frac{\dfrac{T'_1}{\Gamma \to \Delta, s_1 \doteq t_1} \qquad s_1 \doteq t_1, \Gamma \to fs_1...s_n \doteq ft_1...t_n, \Delta}{\Gamma \to fs_1...s_n \doteq ft_1...t_n, \Delta}$$

We finish the proof with one more atomic cut:

$$\frac{\dfrac{\Gamma \to fs_1..., s_n \doteq ft_1...t_n, \Delta}{\Gamma \to \Delta, fs_1..., s_n \doteq ft_1...t_n} \qquad \dfrac{\dfrac{T'_0}{\Gamma, fs_1...s_n \doteq ft_1...t_n \to \Delta}}{fs_1...s_n \doteq ft_1...t_n, \Gamma \to \Delta}}{\Gamma \to \Delta}$$

The trees T'_0, T'_1, ..., T'_n have been obtained using the induction hypothesis. Note that applications of exchange and contraction rules are implicit in the above proofs.

(iii) The root of the $G_=$-proof tree T is labeled with the conclusion of the rule:

$$\frac{\Gamma, ((s_1 \doteq t_1) \wedge ... \wedge (s_n \doteq t_n) \wedge Ps_1...s_n) \supset Pt_1...t_n \to \Delta}{\Gamma \to \Delta}$$

This case is handled as case (ii). This concludes the proof of the theorem. □

Corollary (A version of Gentzen Hauptsatz for LK_e) A sequent is LK_e-provable if and only if it is LK_e-provable without essential cuts.

PROBLEMS

6.3.1. The set of *closed equality axioms* is the set of closed formulae given below:

$$\forall x(x \doteq x)$$

$$\forall x_1 ... \forall x_n \forall y_1 ... \forall y_n ((x_1 \doteq y_1 \land ... \land x_n \doteq y_n) \supset (f(x_1, ..., x_n) \doteq f(y_1, ..., y_n)))$$

$$\forall x_1 ... \forall x_n \forall y_1 ... \forall y_n (((x_1 \doteq y_1 \land ... \land x_n \doteq y_n) \land P(x_1, ..., x_n)) \supset P(y_1, ..., y_n))$$

Prove that the closed equality axioms are LK_e-provable.

6.3.2. Prove that the axioms of definition 6.3.1 are valid.

6.3.4. Finish the proof of the cases in lemma 6.3.1.

6.3.5. Prove case (iii) in the proof of theorem 6.3.1.

6.4 Gentzen's Hauptsatz for Sequents in NNF

Combining ideas from Smullyan and Schwichtenberg (Smullyan, 1968, Barwise, 1977), we formulate a sequent calculus $G1^{nnf}$ in which sequents consist of formulae in negation normal form. Such a system shares characteristics of both G and LK but the main difference is that sequents consist of pairs of *sets* rather than pairs of *sequences*. The main reason for using sets rather than sequences is that the structural rules become unnecessary. As a consequence, an induction argument simpler than Gentzen's original argument (Szabo, 1969) can be used in the proof of the cut elimination theorem. The advantage of considering formulae in negation normal form is that fewer inference rules need to be considered. Since every formula is equivalent to a formula in negation normal form, there is actually no loss of generality.

6.4.1 Negation Normal Form

The definition of a formula in negation normal form given in definition 3.4.8 is extended to the first-order case as follows.

Definition 6.4.1 The set of formulae in *negation normal form* (for short, NNF) is the smallest set of formulae such that

(1) For every atomic formula A, A and $\neg A$ are in NNF;

(2) If A and B are in NNF, then $(A \lor B)$ and $(A \land B)$ are in NNF.

(3) If A is in NNF, then $\forall x A$ and $\exists x A$ are in NNF.

Lemma 6.4.1 Every formula is equivalent to another formula in NNF.

Proof: The proof proceeds by induction on formulae as in lemma 3.4.4. Careful checking of the proof of lemma 3.4.4 reveals that for the propositional connectives, only the case where A is of the form $\neg\forall x B$ or $\neg\exists x B$ needs to be considered. If A is of the form $\neg\forall x B$, by lemma 5.3.6(8) $\neg\forall x B$ is equivalent to $\exists x \neg B$, and since $\neg B$ has fewer connectives than $\neg \forall x B$, by the induction hypothesis $\neg B$ has a NNF B'. By lemma 5.3.7, A is equivalent to $\exists x B'$, which is in NNF. The case where A is of the form $\neg \exists x B$ is similar. Finally, if A is of the form $\forall x B$ or $\exists x B$, by the induction hypothesis B is equivalent to a formula B' is NNF, and by lemma 5.3.7 $\forall x B$ is equivalent to $\forall x B'$ and $\exists x B$ is equivalent to $\exists x B'$. $\square$

EXAMPLE 6.4.1

Let

$$A = \forall x(P(x) \lor \neg \exists y(Q(y) \land R(x, y))) \lor \neg(P(y) \land \neg \forall x P(x))$$

The NNF of $\neg \exists y(Q(y) \land R(x, y))$ is

$$\forall y(\neg Q(y) \lor \neg R(x, y)).$$

The NNF of $\neg(P(y) \land \neg \forall x P(x))$ is

$$(\neg P(y) \lor \forall x P(x)).$$

The NNF of A is

$$\forall x(P(x) \lor \forall y(\neg Q(y) \lor \neg R(x, y))) \lor (\neg P(y) \lor \forall x P(x)).$$

We now define a new Gentzen system in which sequents are pairs of sets of formulae in NNF. First, we treat the case of languages without equality.

6.4.2 The Gentzen System $G1^{nnf}$

The axioms and inference rules of $G1^{nnf}$ are defined as follows.

Definition 6.4.2 The sequents of the system $G1^{nnf}$ are pairs $\Gamma \to \Delta$, where Γ and Δ are finite *sets* of formulae in NNF. Given two sets of formuae Γ and Δ, the expression Γ, Δ denotes the *union* of the sets Γ and Δ, and similarly, if A is a formula, Γ, A denotes the set $\Gamma \cup \{A\}$. The inference rules are the rules listed below:

(1) Cut rule:

$$\frac{\Gamma \to \Delta, A \qquad A, \Lambda \to \Theta}{\Gamma, \Lambda \to \Delta, \Theta}$$

A is called the *cut formula* of this inference.

(2) Propositional logical rules:

$$\frac{A, \Gamma \to \Delta}{A \land B, \Gamma \to \Delta}\ (\land : left) \quad \text{and} \quad \frac{B, \Gamma \to \Delta}{A \land B, \Gamma \to \Delta}\ (\land : left)$$

$$\frac{\Gamma \to \Delta, A \qquad \Gamma \to \Delta, B}{\Gamma \to \Delta, A \wedge B} \ (\wedge : right)$$

$$\frac{A, \Gamma \to \Delta \qquad B, \Gamma \to \Delta}{A \vee B, \Gamma \to \Delta} \ (\vee : left)$$

$$\frac{\Gamma \to \Delta, A}{\Gamma \to \Delta, A \vee B} \ (\vee : right) \quad \text{and} \quad \frac{\Gamma \to \Delta, B}{\Gamma \to \Delta, A \vee B} \ (\vee : right)$$

In the rules above, $A \vee B$ and $A \wedge B$ are called the *principal formulae* and A, B the *side formulae* of the inference.

(3) Quantifier rules

In the quantifier rules below, x is any variable and y is any variable free for x in A and not free in A, unless $y = x$ ($y \notin FV(A) - \{x\}$). The term t is any term free for x in A.

$$\frac{A[t/x], \Gamma \to \Delta}{\forall x A, \Gamma \to \Delta} \ (\forall : left) \qquad \frac{\Gamma \to \Delta, A[y/x]}{\Gamma \to \Delta, \forall x A} \ (\forall : right)$$

$$\frac{A[y/x], \Gamma \to \Delta}{\exists x A, \Gamma \to \Delta} \ (\exists : left) \qquad \frac{\Gamma \to \Delta, A[t/x]}{\Gamma \to \Delta, \exists x A} \ (\exists : right)$$

In both the $(\forall : right)$-rule and the $(\exists : left)$-rule, the variable y does *not* occur free in the lower sequent. In these rules, the variable y is called the *eigenvariable* of the inference. The condition that the eigenvariable does not occur free in the conclusion of the rule is called the *eigenvariable condition*. The formula $\forall x A$ (or $\exists x A$) is called the *principal formula* of the inference, and the formula $A[t/x]$ (or $A[y/x]$) the *side formula* of the inference.

The *axioms* of $G1^{nnf}$ are all sequents of the form

$$\begin{aligned} \Gamma, A &\to A, \Delta, \\ \Gamma, \neg A &\to \neg A, \Delta, \\ \Gamma, A, \neg A &\to \Delta, \text{ or} \\ \Gamma &\to \Delta, A, \neg A, \end{aligned}$$

with A *atomic*.

The notions of *deduction trees* and *proof trees* are defined as usual, but with the rules and axioms of $G1^{nnf}$. It is readily shown that the system $G1^{nnf}$ is sound. We can also prove that $G1^{nnf}$ is complete for sequents in NNF.

6.4.3 Completeness of $G1^{nnf}$

The following lemma is shown using theorem 5.5.1.

Lemma 6.4.2 (Completeness of $G1^{nnf}$) Every valid $G1^{nnf}$-sequent has a $G1^{nnf}$-proof.

Proof: First, we check that the proof given in theorem 5.5.1 can be adapted to hold for sequents consisting of sets rather than sequences. In order to simulate the $\forall : left$ rule and the $\exists : right$ rule of G using the quantifier rules of $G1^{nnf}$, we use the fact that in $G1^{nnf}$, sequents consist of sets. Since $\forall xB, \Gamma \to \Delta$ and $\forall xB, \forall xB, \Gamma \to \Delta$ actually denote the same sequent, we can apply the $\forall : left$ rule (of $G1^{nnf}$) to $\forall xB$, with the formulae in $\forall xB, \Gamma$ and Δ as auxiliary formulae, obtaining:

$$\frac{\forall xB, B[t/x], \Gamma \to \Delta}{\forall xB, \Gamma \to \Delta}$$

The case of $\exists : right$ is similar. We simulate the $\wedge : left$ rule of G and the $\vee : right$ of G as in the proof of theorem 3.6.1. For example, the following derivation simulates $\wedge : left$ of G:

$$\cfrac{\cfrac{A, B, \Gamma \to \Delta}{A \wedge B, B, \Gamma \to \Delta} \quad \wedge : left \text{ applied to } A}{A \wedge B, \Gamma \to \Delta} \quad \wedge : left \text{ applied to } B$$

We obtain proofs in which the conditions for declaring that a sequent is an axiom are as follows: A sequent $\Gamma \to \Delta$ is an axiom iff any of the following conditions holds:

(i) Γ and Δ have some formula A in common; or

(ii) Γ contains some atomic formula B and its negation $\neg B$; or

(iii) Δ contains some atomic formula B and its negation $\neg B$.

However, the formula A may not be a *literal* (that is, an atomic formula, or the negation of an atomic formula). To make sure that in (i) A is a literal, we use the method of lemma 6.3.1, complemented by the cases of quantified formulae. We consider the case in which $A = \forall xB$ is on the left of $\to$, the case $A = \exists xB$ on the right of $\to$ being similar. Assume that the axiom is $\Gamma, \forall xB \to \Delta, \forall xB$. Then we have the following proof:

$$\frac{\dfrac{\Gamma, B[z/x] \to \Delta, B[z/x]}{\Gamma, \forall x B \to \Delta, B[z/x]}}{\Gamma, \forall x B \to \Delta, \forall x B}$$

where z is a new variable.

The top sequent is an axiom, and $B[z/x]$ has fewer connectives than $\forall x B$. We conclude by induction on the number of connectives in $B[z/x]$. The details are left as an exercise. □

6.4.4 The Cut Elimination Theorem for $G1^{nnf}$

We now proceed with the proof of the cut elimination theorem for $G1^{nnf}$. The proof uses a method due to Schwichtenberg (adapted from Tait, Tait, 1968), which consists of a single induction on the cut-rank of a proof with cut. This proof is simpler than Gentzen's original, because the system $G1^{nnf}$ does not have structural rules. This is the reason a simple induction on the cut-rank works (as opposed to the the induction used in Gentzen's original proof, which uses a lexicographic ordering). Furthermore, the proof of the theorem also yields an upper bound on the size of the resulting cut-free proof. The key parameter of the proof, is the cut-rank of a $G1^{nnf}$-proof. First, we need to define the degree of a formula in NNF. Roughly speaking, the degree of a formula in NNF is the depth of the tree representing that formula, ignoring negations.

Definition 6.4.3 The *degree* $|A|$ of a formula A in NNF is defined inductively as follows:

(i) If A is an atomic formula or the negation of an atomic formula, then

$$|A| = 0;$$

(ii) If A is either of the form $(B \vee C)$ or $(B \wedge C)$, then

$$|A| = max(|B|, |C|) + 1;$$

(iii) If A is either of the form $\forall x B$ or $\exists x B$, then

$$|A| = |B| + 1.$$

The cut-rank is defined as follows.

Definition 6.4.4 Let T be a $G1^{nnf}$-proof. The *cut-rank* $c(T)$ of T is defined inductively as follows. If T is an axiom, then $c(T) = 0$. If T is not an axiom, the last inference has either one or two premises. In the first case, the premise of that inference is the root of a subtree T_1. In the second case, the left premise is the root of a subtree T_1, and the right premise is the root of a subtree T_2. If the last inference is not a cut, then if it has a single premise,

$$c(T) = c(T_1),$$

else

$$c(T) = max(c(T_1), c(T_2)).$$

If the last inference is a cut with cut formula A, then

$$c(T) = max(|A| + 1, c(T_1), c(T_2)).$$

Note that $c(T) = 0$ iff T is cut free. We will need a number of lemmas to establish the cut elimination theorem for $G1^{nnf}$. In some of these proofs, it will be necessary to replace in a proof all free occurrences of variable y by a new variable z not occurring in the proof. This substitution process is defined as follows.

Definition 6.4.5 Given a formula A, we say that the variable y is *not bound in the formula* A iff $y \notin BV(A)$. The variable y is *not bound in the sequent* $\Gamma \to \Delta$ iff y is not bound in any formula in Γ or Δ. The variable y is *not bound in the deduction tree* T iff it is not bound in any sequent occurring in T. Given a formula A and two variables y, z, the formula $A[z/y]$ is defined as in definition 5.2.6. For a sequent $\Gamma \to \Delta$, the sequent $(\Gamma \to \Delta)[z/y]$ is the sequent obtained by substituting z for y in all formulae in $\Gamma \to \Delta$. For a deduction tree T, a variable y not bound in T, and a variable z not occurring in T, the deduction tree $T[z/y]$ is the result of replacing every sequent $\Gamma \to \Delta$ in T by $(\Gamma \to \Delta)[z/y]$. This operation can be defined more precisely by induction on proof trees, the simple details being left to the reader.

A similar definition can be given for the result $T[t/y]$ of substituting a term t for a variable y not bound in T, provided that t is free for y in every formula in which it is substituted, and that y and the variables in $FV(t)$ are distinct from all eigenvariables in T. In order to justify that $T[z/y]$ are $T[t/y]$ are indeed proof trees when T is, the following technical lemma is needed.

Lemma 6.4.3 Let $\Gamma \to \Delta$ be a sequent and T an $G1^{nnf}$-proof for $\Gamma \to \Delta$. Assume that y is any variable not bound in the proof tree T.

(i) For any variable z not occurring in T, the result $T[z/y]$ of substituting z for all occurrences of y in T is a proof tree for $\Gamma[z/y] \to \Delta[z/y]$.

(ii) If t is a term free for y in every formula in which it is substituted, and y and the variables in $FV(t)$ are distinct from all eigenvariables in T, then $T[t/y]$ is a proof tree for $\Gamma[t/y] \to \Delta[t/y]$.

Proof: We proceed by induction on proof trees. We only treat some key cases, leaving the others as an exercise. We consider (i). If T is an axiom

$\Gamma \to \Delta$, it is clear that $(\Gamma \to \Delta)[z/y]$ is also an axiom. The propositional rules present no difficulty and are left to the reader. We consider two of the quantifier rules: $\forall : left$ and $\forall : right$.

Case 1: The bottom inference is $\forall : left$:

$$\frac{\dfrac{T_1}{A[t/x], \Gamma \to \Delta}}{\forall x A, \Gamma \to \Delta}$$

where t is free for x in A.

By the induction hypothesis, $T_1[z/y]$ is a proof tree of $A[t/x][z/y], \Gamma[z/y] \to \Delta[z/y]$. Since we have assumed that y is not bound in the proof T, $y \neq x$. Hence, $(\forall x A)[z/y] = \forall x A[z/y]$. By the result of problem 5.2.7,

$$A[t/x][z/y] = A[z/y][t[z/y]/x],$$

and since z does not occur in T, $t[z/y]$ is free for x in $A[z/y]$. But then, $T[z/y]$ is the proof tree:

$$\frac{\dfrac{T_1[z/y]}{A[z/y][t[z/y]/x], \Gamma[z/y] \to \Delta[z/y]}}{\forall x A[z/y], \Gamma[z/y] \to \Delta[z/y]}$$

Case 2: The bottom inference is $\forall : right$:

$$\frac{\dfrac{T_1}{\Gamma \to \Delta, A[w/x]}}{\Gamma \to \Delta, \forall x A}$$

where w is not free in $\Gamma \to \Delta, \forall x A$.

There are two subcases.

Subcase 2.1: If $y = w$, since w does not occur free in $\Gamma \to \Delta, \forall x A$, the variable w is not free in $\forall x A$, Γ or Δ. By the induction hypothesis, $T_1[z/y]$ is a proof tree for

$$(\Gamma \to \Delta, A[y/x])[z/y] = \Gamma \to \Delta, A[z/x].$$

Also, since z does not occur in T, z does not occur in $\Gamma \to \Delta, \forall x A$, the $\forall : right$ rule is applicable and $T[z/y]$ is a proof tree.

Subcase 2.2: $y \neq w$. By the induction hypothesis, $T_1[z/y]$ is a proof tree for $\Gamma[z/y] \to \Delta[z/y], A[w/x][z/y]$. By problem 5.2.7, we have

$$A[w/x][z/y] = A[z/y][w[z/y]/x],$$

but since $w \neq y$,

$$A[w/x][z/y] = A[z/y][w/x].$$

Also, since z does not occur in T, $z \neq w$, and so w does not occur free in $\Gamma[z/y] \rightarrow \Delta[z/y], \forall x A[z/y]$. Hence, the $\forall : right$ rule is applicable and $T[z/y]$ is a proof tree:

$$\cfrac{\cfrac{T_1[z/y]}{\Gamma[z/y] \rightarrow \Delta[z/y], A[z/y][w/x]}}{\Gamma[z/y] \rightarrow \Delta[z/y], \forall x A[z/y]}$$

It should be noted that in the proof of (ii), the condition that y is distinct from all eigenvariables in T rules out subcase 2.1. □

Lemma 6.4.4 (Substitution lemma) Let T be a $G1^{nnf}$-proof of a sequent $\Gamma \rightarrow \Delta$ such that the variable x is not bound in T. For any term t free for x in $\Gamma \rightarrow \Delta$, a proof $T'(t)$ of $\Gamma[t/x] \rightarrow \Delta[t/x]$ can be constructed such that $T'(t)$ and T have same depth, x and the variables in $FV(t)$ are distinct from all eigenvariables in $T'(t)$ and $c(T) = c(T'(t))$.

Proof: By induction on proof trees using lemma 6.4.3. For a similar proof, see lemma 7.3.1. □

Lemma 6.4.5 (Weakening lemma) Given a $G1^{nnf}$-proof T of a sequent $\Gamma \rightarrow \Delta$, for any formula A (in NNF), a proof T' of $A, \Gamma \rightarrow \Delta$ (resp. $\Gamma \rightarrow \Delta, A$) can be obtained such that T and T' have the same depth, all variables free in A are distinct from all eigenvariables in T and $c(T') = c(T)$.

Proof: Straightforward induction on proof trees, similar to that of lemma 6.4.3. □

The proof T' given by the weakening lemma will also be denoted by (T, A).

Lemma 6.4.6 (Inversion lemma) (i) If a sequent $\Gamma \rightarrow \Delta, A \wedge B$ has a $G1^{nnf}$-proof T (resp. $A \vee B, \Gamma \rightarrow \Delta$ has a $G1^{nnf}$-proof T), a $G1^{nnf}$-proof T_1 of $\Gamma \rightarrow \Delta, A$ and a $G1^{nnf}$-proof T_2 of $\Gamma \rightarrow \Delta, B$ can be constructed (resp. a $G1^{nnf}$-proof T_1 of $A, \Gamma \rightarrow \Delta$ and a $G1^{nnf}$-proof T_2 of $B, \Gamma \rightarrow \Delta$ can be constructed) such that, $depth(T_1), depth(T_2) \leq depth(T)$ and $c(T_1), c(T_2) \leq c(T)$.

(ii) If $\Gamma \rightarrow \Delta, \forall x B$ has a $G1^{nnf}$-proof T (resp. $\exists x B, \Gamma \rightarrow \Delta$ has a $G1^{nnf}$-proof T), then for any variable y not bound in T and distinct from all eigenvariables in T, a $G1^{nnf}$-proof T_1 of $\Gamma \rightarrow \Delta, B[y/x]$ can be constructed (resp. a $G1^{nnf}$-proof T_1 of $B[y/x], \Gamma \rightarrow \Delta$ can be constructed), such that $depth(T_1) \leq depth(T)$ and $c(T_1) \leq c(T)$.

Proof: The proofs of (i) and (ii) are similar, both by induction on proof trees. We consider (ii), leaving (i) as an exercise.

If $\forall xB$ belongs to Δ, the result follows by the weakening lemma (lemma 6.4.5), using the proof $(T, B[y/x])$ given by that lemma. If $\forall xB$ does not belong to Δ, there are two cases.

Case 1: $\forall xB$ is not the principal formula of the last inference. There are several subcases depending on that inference. Let us consider the $\wedge : right$ rule, the other subcases being similar and left as an exercise. The proof has the form

$$\dfrac{\dfrac{S_1}{\Gamma \to \Delta, \forall xB, C} \qquad \dfrac{S_2}{\Gamma \to \Delta, \forall xB, D}}{\Gamma \to \Delta, \forall xB, C \wedge D}$$

By the induction hypothesis, for any variable y not bound in S_1 or S_2 and distinct from all eigenvariables in S_1 or S_2, we can find proofs T_1 for $\Gamma \to \Delta, B[y/x], C$ and T_2 for $\Gamma \to \Delta, B[y/x], D$, such that $depth(T_i) \leq depth(S_i)$ and $c(T_i) \leq c(S_i)$, for $i = 1, 2$. We conclude using the following proof:

$$\dfrac{\dfrac{T_1}{\Gamma \to \Delta, B[y/x], C} \qquad \dfrac{T_2}{\Gamma \to \Delta, B[y/x], D}}{\Gamma \to \Delta, B[y/x], C \wedge D}$$

Case 2: $\forall xB$ is the principal formula of the last inference. Using the weakening lemma, we can make sure that the last inference is of the form

$$\dfrac{\dfrac{T_1}{\Gamma \to \Delta, \forall xB, B[y/x]}}{\Gamma \to \Delta, \forall xB}$$

replacing the proof T by $(T, \forall xB)$ if necessary. Using lemma 6.4.3, we can also make sure that y is not bound in T_1 (or T) and is distinct from all eigenvariables in T_1 (and T). Then, the induction hypothesis applies to the variable y in the lower sequent of the proof T_1, and we can find a proof T_1' of $\Gamma \to \Delta, B[y/x]$ such that $depth(T_1) < depth(T)$ and $c(T_1) \leq c(T)$. Using lemma 6.4.3 again, we actually have a proof of $\Gamma \to \Delta, B[z/x]$ for any variable z not bound in T_1' and distinct from all eigenvariables in T_1', establishing (ii). The proof of the other cases is similar. $\square$

We are now ready for the main lemma which, shows how cuts are eliminated.

Lemma 6.4.7 (Reduction lemma for $G1^{nnf}$) Let T_1 be a $G1^{nnf}$-proof of $\Gamma \to \Delta, A$, and T_2 a $G1^{nnf}$-proof of $A, \Lambda \to \Theta$, and assume that

$$c(T_1), c(T_2) \leq |A|.$$

A $G1^{nnf}$-proof T of

$$\Gamma, \Lambda \rightarrow \Delta, \Theta$$

can be constructed, such that

$$depth(T) \leq depth(T_1) + depth(T_2) \quad \text{and} \quad c(T) \leq |A|.$$

Proof: We proceed by induction on $depth(T_1) + depth(T_2)$.

Case 1: Either A is not the principal formula of the last inference of T_1, or A is not the principal formula of the last inference of T_2. By symmetry, we can assume the former. There are several subcases, depending on the last inference. Let us consider the $\wedge : right$ rule, the other subcases being similar and left as an exercise. The proof has the form

$$\frac{\dfrac{S_1}{\Gamma \rightarrow \Delta', C, A} \qquad \dfrac{S_2}{\Gamma \rightarrow \Delta', D, A}}{\Gamma \rightarrow \Delta', C \wedge D, A}$$

By the induction hypothesis, we can find proofs T_1' for $\Gamma, \Lambda \rightarrow \Delta', \Theta, C$ and T_2' for $\Gamma, \Lambda \rightarrow \Delta', \Theta, D$, such that $depth(T_i') < depth(T_1) + depth(T_2)$ and $c(T_i') \leq |A|$, for i=1,2. The result follows by the inference

$$\frac{\Gamma, \Lambda \rightarrow \Delta', \Theta, C \qquad \Gamma, \Lambda \rightarrow \Delta', \Theta, D}{\Gamma, \Lambda \rightarrow \Delta', C \wedge D, \Theta}$$

Case 2: A is the principal formula of the last inference of both T_1 and T_2.

Case 2.1: A is a literal; that is, an atomic formula, or the negation of an atomic formula. In this case, both $\Gamma \rightarrow \Delta, A$ and $A, \Lambda \rightarrow \Theta$ are axioms. By considering all possible cases, it can be verified that $\Gamma, \Lambda \rightarrow \Delta, \Theta$ is an axiom. For example, if A is atomic, Δ contains $\neg A$ and Λ contains $\neg A$, $\Gamma, \Lambda \rightarrow \Delta, \Theta$ is an axiom.

Case 2.2: A is of the form $(B \vee C)$. Using the weakening lemma, we can make sure that the last inference of T_1 is of the form

$$\frac{\dfrac{S_0}{\Gamma \rightarrow \Delta, A, B}}{\Gamma \rightarrow \Delta, A}$$

or

$$\dfrac{\dfrac{S_0}{\Gamma \rightarrow \Delta, A, C}}{\Gamma \rightarrow \Delta, A}$$

replacing T_1 by (T_1, A) if necessary. Consider the first case, the other being similar. By the induction hypothesis, we can find a proof T_1' for $\Gamma, \Lambda \rightarrow \Delta, \Theta, B$, such that $depth(T_1') < depth(T_1) + depth(T_2)$ and $c(T_1') \leq |A|$. By the inversion lemma, we can find a proof T_2' of $B, \Lambda \rightarrow \Theta$ such that $depth(T_2') \leq depth(T_2)$ and $c(T_2') \leq |A|$. The following proof T'

$$\dfrac{\dfrac{T_1'}{\Gamma, \Lambda \rightarrow \Delta, \Theta, B} \qquad \dfrac{T_2'}{B, \Lambda \rightarrow \Theta}}{\Gamma, \Lambda \rightarrow \Delta, \Theta}$$

is such that $depth(T') \leq depth(T_1) + depth(T_2)$ and has cut-rank $c(T') \leq |A|$, since $|B| < |A|$.

Case 2.3: A is of the form $(B \wedge C)$. This case is symmetric to case 2.2.

Case 2.4: A is of the form $\exists x B$. As in case 2.2, we can assume that A belongs to the premise of the last inference in T_1, so that T_1 is of the form

$$\dfrac{\dfrac{S_0}{\Gamma \rightarrow \Delta, A, B[t/x]}}{\Gamma \rightarrow \Delta, A}$$

By the induction hypothesis, we can find a proof tree T_1' for $\Gamma, \Lambda \rightarrow \Delta, \Theta, B[t/x]$, such that $depth(T_1') < depth(T_1) + depth(T_2)$ and $c(T_1') \leq |A|$. By the inversion lemma, for any variable y not bound in T_2 and distinct from all eigenvariables in T_2, there is a proof T_2' for $B[y/x], \Lambda \rightarrow \Theta$, such that $depth(T_2') \leq depth(T_2)$ and $c(T_2') \leq |A|$. By the substitution lemma, we can construct a proof T_2'' for $B[t/x], \Lambda \rightarrow \Theta$, also such that $depth(T_2'') \leq depth(T_2)$ and $c(T_2'') \leq |A|$. Since $|B[t/x]| < |A|$, the proof obtained from T_1' and T_2'' by applying a cut to $B[t/x]$ has cut-rank $\leq |A|$.

Case 2.5: A is of the form $\forall x B$. This case is symmetric to case 2.4. This concludes all the cases. □

Finally, we can prove the cut elimination theorem.

The function $exp(m, n, p)$ is defined recursively as follows:

$$exp(m, 0, p) = p;$$
$$exp(m, n+1, p) = m^{exp(m,n,p)}.$$

This function grows extremely fast in the argument n. Indeed, $exp(m, 1, p) = m^p$, $exp(m, 2, p) = m^{m^p}$, and in general, $exp(m, n, p)$ is an iterated stack of exponentials of height n, topped with a p:

$$exp(m, n, p) = \left. m^{m^{m^{\cdot^{\cdot^{\cdot^{m^p}}}}}} \right\} n$$

The Tait-Schwichtenberg's version of the cut elimination theorem for $G1^{nnf}$ follows.

Theorem 6.4.1 (Cut elimination theorem for $G1^{nnf}$) Let T be a $G1^{nnf}$-proof with cut-rank $c(T)$ of a sequent $\Gamma \rightarrow \Delta$. A cut-free proof T^* for $\Gamma \rightarrow \Delta$ such that $depth(T^*) \leq exp(2, c(T), depth(T))$ can be constructed.

Proof: We prove the following claim by induction on the depth of proof trees.

Claim: Let T be a $G1^{nnf}$-proof with cut-rank $c(T)$ for a sequent $\Gamma \rightarrow \Delta$. If $c(T) > 0$ then we can construct a proof T' for $\Gamma \rightarrow \Delta$, such that

$$c(T') < c(T) \quad \text{and} \quad depth(T') \leq 2^{depth(T)}.$$

Proof of Claim: If either the last inference of T is not a cut, or it is a cut and $c(T) > |A| + 1$, where A is the cut formula of the last inference, we can apply the induction hypothesis to the immediate subtrees T_1 or T_2 (or T_1) of T. We are left with the case in which the last inference is a cut and $c(T) = |A| + 1$. The proof is of the form

$$\dfrac{\dfrac{T_1}{\Gamma \rightarrow \Delta, A} \qquad \dfrac{T_2}{A, \Gamma \rightarrow \Delta}}{\Gamma \rightarrow \Delta}$$

By the induction hypothesis, we can construct a proof T_1' for $\Gamma \rightarrow \Delta, A$ and a proof T_2' for $A, \Gamma \rightarrow \Delta$, such that $c(T_i') \leq |A|$ and $depth(T_i') \leq 2^{depth(T_i)}$, for $i = 1, 2$. Applying the reduction lemma, we obtain a proof T' such that, $c(T') \leq |A|$ and $depth(T') \leq depth(T_1') + depth(T_2')$. But

$$depth(T_1') + depth(T_2') \leq 2^{depth(T_1)} + 2^{depth(T_2)} \leq$$
$$2^{max(depth(T_1), depth(T_2))+1} = 2^{depth(T)}.$$

Hence, the claim holds for T'. □

The proof of the theorem follows by induction on $c(T)$, and by the definition of $exp(2, m, n)$. □

It is remarkable that theorem 6.4.1 provides an upper bound on the depth of cut-free proofs obtained by converting a proof with cuts. Note that the "blow up" in the size of the proof can be very large.

Cut-free proofs are "direct" (or analytic), in the sense that all inferences are purely mechanical, and thus require no ingenuity. Proofs with cuts are "indirect" (or nonanalytic), in the sense that the cut formula in a cut rule may not be a subformula of any of the formulae in the conclusion of the inference. Theorem 6.4.1 suggests that if some ingenuity is exercised in constructing proofs with cuts, the size of a proof can be reduced significantly. It gives a measure of the complexity of proofs. Without being very rigorous, we can say that theorem 6.4.1 suggests that there are theorems that have no easy proofs, in the sense that if the steps are straightforward, the proof is very long, or else if the proof is short, the cuts are very ingenious. Such an example is given in Statman, 1979.

We now add equality axioms to the system $G1^{nnf}$.

6.4.5 The System $G1^{nnf}_{=}$

The axioms and inference rules of the system $G1^{nnf}_{=}$ are defined as follows.

Definition 6.4.6 The system $G1^{nnf}_{=}$ is obtained by adding to $G1^{nnf}$ the following sequents as axioms. All sequents of the form

(i) $\Gamma \to \Delta, t \doteq t$;

(ii) For every n-ary function symbol f,

$$\Gamma, s_1 \doteq t_1, s_2 \doteq t_2, ..., s_n \doteq t_n \to \Delta, fs_1...s_n \doteq ft_1...t_n$$

(iii) For every n-ary predicate symbol P (including $\doteq$),

$$\Gamma, s_1 \doteq t_1, s_2 \doteq t_2, ..., s_n \doteq t_n, Ps_1...s_n \to \Delta, Pt_1...t_n$$

and all sequents obtained from the above sequents by applications of $\neg : left$ and $\neg : right$ rules to the atomic formulae $t \doteq t$, $s_i \doteq t_i$, $fs_1...s_n \doteq ft_1...t_n$, $Ps_1...s_n$ and $Pt_1...t_n$.

For example, $\to \neg(s \doteq t), f(s) \doteq f(t)$ is an equality axiom. It is obvious that these sequents are valid and that the system $G1^{nnf}_{=}$ is sound.

Lemma 6.4.8 (Completeness of $G1^{nnf}_{=}$) For the system $G1^{nnf}_{=}$, every valid sequent is provable.

Proof: The lemma can be proved using the completeness of LK_e (theorem 6.3.1). First, we can show that in an LK_e-proof, all weakenings can be moved above all other inferences. Then, we can show how such a proof can be simulated by a $G1^{nnf}_{=}$-proof. The details are rather straighforward and are left as an exercise. □

6.4.6 The Cut Elimination Theorem for $G1^{nnf}_{=}$

Gentzen's cut elimination theorem also holds for $G1^{nnf}_{=}$ if an *inessential cut* is defined as a cut in which the cut formula is a *literal* B or $\neg B$, where B is *equation* $s \doteq t$ (but not an atomic formula of the form $Ps_1...s_n$, where P is a predicate symbol different from $\doteq$). This has interesting applications, such as the completeness of equational logic (see the problems). The proof requires a modification of lemma 6.4.6. The *cut-rank* of a proof is now defined by considering essential cuts only.

Lemma 6.4.9 (Reduction lemma for $G1^{nnf}_{=}$) Let T_1 be a $G1^{nnf}_{=}$-proof of $\Gamma \to \Delta, A$, and T_2 a $G1^{nnf}_{=}$-proof of $A, \Lambda \to \Theta$, and assume that $c(T_1), c(T_2) \leq |A|$. Let m be the maximal rank of all predicate symbols P such that some literal $Pt_1...t_n$ or $\neg Pt_1...t_n$ is a cut formula in either T_1 or T_2. A $G1^{nnf}_{=}$-proof T of $\Gamma, \Lambda \to \Delta, \Theta$ can be constructed such that

$$c(T) \leq |A| \quad \text{and} \quad depth(T) \leq depth(T_1) + depth(T_2) + m.$$

Proof: We proceed by induction on $depth(T_1) + depth(T_2)$. The only new case is the case in which A is a literal of the form $Pt_1...t_n$ or $\neg Pt_1...t_n$, and one of the two axioms $\Gamma \to \Delta, A$ and $A, \Lambda \to \Theta$ is an equality axiom. Assume that $A = Pt_1...t_n$, the other case being similar. If $\Gamma \to \Delta$ or $\Lambda \to \Theta$ is an axiom, then $\Gamma, \Lambda \to \Delta, \Theta$ is an axiom. Otherwise, we have three cases.

Case 1: $\Gamma \to \Delta, A$ is an equality axiom, but $A, \Lambda \to \Theta$ is not. Then, either $\neg A$ is in Λ or A is in Θ, and $\Gamma \to \Delta, A$ is obtained from some equality axiom of the form

$$\Gamma', s_1 \doteq t_1, s_2 \doteq t_2, ..., s_n \doteq t_n, Ps_1...s_n \to \Delta', Pt_1...t_n$$

by application of $\neg$: *rules*. Since either $Pt_1...t_n$ is in Θ or $\neg Pt_1...t_n$ is in Λ, the sequent $\Gamma, \Lambda \to \Delta, \Theta$ is an equality axiom also obtained from some sequent of the form

$$\Gamma'', s_1 \doteq t_1, s_2 \doteq t_2, ..., s_n \doteq t_n, Ps_1...s_n \to \Delta'', Pt_1...t_n.$$

Case 2: $\Gamma \to \Delta, A$ is not an equality axiom, but $A, \Lambda \to \Theta$ is. This is similar to case 1.

Case 3: Both $\Gamma \to \Delta, A$ and $A, \Lambda \to \Theta$ are equality axioms. In this case, $\Gamma \to \Delta, A$ is obtained from some equality axiom of the form

$$\Gamma', s_1 \doteq t_1, s_2 \doteq t_2, ..., s_n \doteq t_n, Ps_1...s_n \to \Delta', Pt_1...t_n$$

by application of $\neg$: *rules*, and $A, \Lambda \to \Theta$ is obtained from some equality axiom of the form

$$\Lambda', t_1 \doteq r_1, t_2 \doteq r_2, ..., t_n \doteq r_n, Pt_1...t_n \to \Theta', Pr_1...r_n$$

by applications of $\neg : rules$.

Then, $\Gamma, \Lambda \rightarrow \Delta, \Theta$ is obtained by applying negation rules to the sequent

$$\Gamma', \Lambda', s_1 \doteq t_1, s_2 \doteq t_2, ..., s_n \doteq t_n, t_1 \doteq r_1, t_2 \doteq r_2, ..., t_n \doteq r_n, Ps_1...s_n \\ \rightarrow \Delta', \Theta', Pr_1...r_n.$$

A proof with only inessential cuts can be given using axioms derived by applying $\neg : rules$ to the provable (with no essential cuts) sequents

$$s_i \doteq t_i, t_i \doteq r_i \rightarrow s_i \doteq r_i,$$

for $i = 1, ..., n$, and the axiom

$$\Gamma', \Lambda', s_1 \doteq r_1, ..., s_n \doteq r_n, Ps_1...s_n \rightarrow \Delta', \Theta', Pr_1...r_n,$$

and n inessential cuts (the i-th cut with cut formula $s_i \doteq r_i$ or $\neg s_i \doteq r_i$). The depth of this proof is n, which is bounded by m. The rest of the proof is left as an exercise. □

We obtain the following cut elimination theorem.

Theorem 6.4.2 (Cut elimination theorem for $G1^{nnf}_{=}$) Let T be a $G1^{nnf}_{=}$-proof with cut-rank $c(T)$ for a sequent $\Gamma \rightarrow \Delta$, and let m be defined as in lemma 6.4.9. A $G1^{nnf}_{=}$-proof T^* for $\Gamma \rightarrow \Delta$ without essential cuts such that $depth(T^*) \leq exp(m+2, c(T), depth(T))$ can be constructed.

Proof: The following claim can be shown by induction on the depth of proof trees:

Claim: Let T be a $G1^{nnf}_{=}$-proof with cut-rank $c(T)$ for a sequent $\Gamma \rightarrow \Delta$. If $c(T) > 0$ then we can construct a $G1^{nnf}_{=}$-proof T' for $\Gamma \rightarrow \Delta$, such that

$$c(T') < c(T) \quad \text{and} \quad depth(T') \leq (m+2)^{depth(T)}.$$

The details are left as an exercise. □

Note: The cut elimination theorem with inessential cuts (with atomic cut formulae of the form $s \doteq t$) also holds for LK_e. The interested reader is refered to Takeuti, 1975.

As an application of theorem 6.3.1, we shall prove Craig's interpolation theorem in the next section. The significance of a proof of Craig's theorem using theorem 6.3.1 is that an interpolant can actually be constructed. In turn, Craig's interpolation theorem implies two other important results: Beth's definability theorem, and Robinson's joint consistency theorem.

PROBLEMS

6.4.1. Convert the following formulae to NNF:

$$(\neg\forall x P(x,y) \lor \forall x R(x,y))$$
$$\forall x(P(x) \supset \neg\exists y R(x,y))$$
$$(\neg\forall x\neg\forall y\neg\forall z P(x,y) \lor \neg\exists x\neg\exists y(\neg\exists z Q(x,y,z) \supset R(x,y)))$$

6.4.2. Convert the following formulae to NNF:

$$(\exists x\forall y P(x,y) \land \forall y\exists x P(y,x))$$
$$(\neg(\forall x P(x) \lor \exists y\neg Q(y)) \lor (\forall z G(z) \lor \exists w\neg Q(w)))$$
$$(\neg\forall x(P(x) \lor \exists y\neg Q(y)) \lor (\forall z P(z) \lor \exists w\neg Q(w)))$$

6.4.3. Write a computer program for converting a formula to NNF.

6.4.4. Give the details of the proof of lemma 6.4.2.

6.4.5. Finish the proof of the cases that have been left out in the proof of lemma 6.4.3.

6.4.6. Prove lemma 6.4.4.

6.4.7. Prove lemma 6.4.5.

6.4.8. Finish the proof of the cases that have been left out in the proof of lemma 6.4.6.

6.4.9. Finish the proof of the cases that have been left out in the proof of lemma 6.4.7.

6.4.10. Prove that for any LK_e-proof, all weakenings can be moved above all other kinds of inferences. Use this result to prove lemma 6.4.8.

6.4.11. Finish the proof of the cases that have been left out in the proof of lemma 6.4.9.

6.4.12. Give the details of the proof of theorem 6.4.2.

6.4.13. Given a set S of formulae, let $Des(S)$ be the set of *immediate descendants* of formulae in S as defined in problem 5.5.18, and define S^n by induction as follows:

$$S^0 = S;$$
$$S^{n+1} = Des(S^n).$$
$$\text{Let } S^* = \bigcup_{n\geq 0} S^n.$$

S^* is called the set of descendants of S.

(a) Prove that for every deduction tree without essential cuts for a sequent $\Gamma_0 \to \Delta_0$, the formulae in the sets of formulae $\Gamma \cup \{\neg B \mid B \in \Delta\}$ for all sequents $\Gamma \to \Delta$ occurring in that tree, belong to S^* or are equations of the form $s \doteq t$, where $S = \Gamma_0 \cup \{\neg B \mid B \in \Delta_0\}$.

(b) Deduce from (a) that not all $G1^{nnf}_{=}$-formulae are provable.

6.4.14. Let $\mathbf{L}$ be a first-order language with equality and with function symbols and constant symbols, but no predicate symbols. Such a language will be called *equational*.

Let $e_1, ..., e_m \to e$ be a sequent where each e_i is a closed formula of the form $\forall x_1 ... \forall x_n (s \doteq t)$, called a *universal equation*, where $\{x_1, ..., x_n\}$ is the set of variables free in $s \doteq t$, and e is an atomic formula of the form $s \doteq t$, called an *equation*. Using theorem 6.4.2, prove that if $e_1, ..., e_n \to e$ is $G1^{nnf}_{=}$-provable, then it is provable using only axioms (including the equality axioms), the cut rule applied to equations (of the form $s \doteq t$), weakenings, and the $\forall : left$ rule.

$*$ **6.4.15.** Let $\mathbf{L}$ be an equational language as defined in problem 6.4.14. A *substitution function* (for short, a *substitution*) is any function $\sigma : \mathbf{V} \to TERM_{\mathbf{L}}$ assigning terms to the variables in $\mathbf{V}$. By theorem 2.4.1, there is a unique homomorphism $\widehat{\sigma} : TERM_{\mathbf{L}} \to TERM_{\mathbf{L}}$ extending σ and defined recursively as follows:

For every variable $x \in \mathbf{V}$, $\widehat{\sigma}(x) = s(x)$.

For every constant c, $\widehat{\sigma}(c) = c$.

For every term $ft_1 ... t_n \in TERM_{\mathbf{L}}$,

$$\widehat{\sigma}(ft_1 ... t_n) = f\widehat{\sigma}(t_1) ... \widehat{\sigma}(t_n).$$

By abuse of language and notation, the function $\widehat{\sigma}$ will also be called a substitution, and will often be denoted by σ.

The subset X of $\mathbf{V}$ consisting of the variables such that $s(x) \neq x$ is called the *support of the substitution*. In what follows, we will be dealing with substitutions of finite support. If a substitution σ has finite support $\{y_1, ..., y_n\}$ and $\sigma(y_i) = s_i$, for $i = 1, .., n$, for any term t, the substitution instance $\widehat{\sigma}(t)$ is also denoted as $t[s_1/y_1, ..., s_n/y_n]$.

Let $E =< e_1, ..., e_n >$ be a sequence of equations, and E' the sequence of their universal closures. (Recall that for a formula A with set of free variables $FV(A) = \{x_1, ..., x_n\}$, $\forall x_1 ... \forall x_n A$ is the *universal closure* of A.)

We define the relation $\longrightarrow_E$ on the set $TERM_{\mathbf{L}}$ of terms as follows. For any two terms t_1, t_2,

$$t_1 \longrightarrow_E t_2$$

iff there is some term r, some equation $s \doteq t \in E$, some substitution σ with support $FV(s) \cup FV(t)$, and some tree address u in r, such that,

$$t_1 = r[u \leftarrow \widehat{\sigma}(s)], \text{ and } t_2 = r[u \leftarrow \widehat{\sigma}(t)].$$

When $t_1 \longrightarrow_E t_2$, we say that t_1 *rewrites* to t_2. In words, t_1 rewrites to t_2 iff t_2 is obtained from t_1 by finding a subterm $\widehat{\sigma}(s)$ of t_1 (called a pattern) which is a substitution instance of the left hand side s of an equation $s \doteq t \in E$, and replacing this subterm by the subterm $\widehat{\sigma}(t)$ obtained by applying the same substitution σ to the right hand side t of the equation.

Let $\longleftrightarrow_E$ be the relation defined such that

$$t_1 \longleftrightarrow_E t_2 \quad \text{iff}$$
$$\text{either } t_1 \longrightarrow_E t_2 \quad \text{or} \quad t_2 \longrightarrow_E t_1,$$

and let $\overset{*}{\longleftrightarrow}_E$ be the reflexive and transitive closure of $\longleftrightarrow_E$.

Our goal is to prove that for every sequence $E =< e_1, ..., e_n >$ of equations and for every equation $s \doteq t$, if E' is the sequence of universal closures of equations in E, then

$$E' \rightarrow s \doteq t \quad \text{is } G1^{nnf}_{=}\text{-provable iff} \quad s \overset{*}{\longleftrightarrow}_E t.$$

During the proof that proceeds by induction, we will have to consider formulae which are universally quantified equations of the form

$$\forall y_1 ... \forall y_m (s \doteq t),$$

where $\{y_1, ..., y_m\}$ is a subset of $FV(s) \cup FV(t)$, including the case $m = 0$ which corresponds to the quantifier-free equation $s \doteq t$. Hence, we will prove two facts.

For every sequence T consisting of (partially) universally quantified equations from E,

(1) If $T \rightarrow s \doteq t$ is $G1^{nnf}_{=}$-provable then $s \overset{*}{\longleftrightarrow}_E t$.

(2) If $s \overset{*}{\longleftrightarrow}_E t$ then $E' \rightarrow s \doteq t$ is $G1^{nnf}_{=}$-provable,

where E' is the universal closure of E.

This is done as follows:

(a) Prove that if $s \overset{*}{\longleftrightarrow}_E t$, then the sequent $E' \rightarrow s \doteq t$ is $G1^{nnf}_{=}$-provable, where E' is the universal closure of E.

Hint: Use induction on the structure of the term r in the definition of $\overset{*}{\longleftrightarrow}_E$.

(b) Given a set E of equations and any equation $v \doteq w$, let $\{E, v \doteq w\}$ denote the union of E and $\{v \doteq w\}$.

Prove that

(i) If $s_i \overset{*}{\longleftrightarrow}_E t_i$, for $1 \leq i \leq n$, then

$$f(t_1, ..., t_n) \overset{*}{\longleftrightarrow}_E f(s_1, ..., s_n).$$

(ii) If $t_1 \overset{*}{\longleftrightarrow}_E t_2$, then for every substitution σ with support $FV(t_1) \cup FV(t_2)$,

$$\widehat{\sigma}(t_1) \overset{*}{\longleftrightarrow}_E \widehat{\sigma}(t_2).$$

(iii) If $v \overset{*}{\longleftrightarrow}_E w$ and $s \overset{*}{\longleftrightarrow}_{\{E, v \doteq w\}} t$, then $s \overset{*}{\longleftrightarrow}_E t$.

Hint: Show that every rewrite step involving $v \doteq w$ as an equation can be simulated using the steps in $v \overset{*}{\longleftrightarrow}_E w$.

(c) Prove that, for every sequence T consisting of (partially) universally quantified equations from E, for any equation $s \doteq t$, if the sequent $T \rightarrow s \doteq t$ is $G1^{nnf}_{=}$-provable then $s \overset{*}{\longleftrightarrow}_E t$.

Hint: Proceed by induction on $G1^{nnf}_{=}$ proofs without essential cuts. One of the cases is that of an inessential cut. If the bottom inference is a cut, it must be of the form

$$\dfrac{\dfrac{S_1}{T_1 \rightarrow v \doteq w} \qquad \dfrac{S_2}{v \doteq w, T_2 \rightarrow s \doteq t}}{T \rightarrow s \doteq t}$$

where T_1 and T_2 are subsets of T. By the induction hypothesis, $v \overset{*}{\longleftrightarrow}_E w$ and $s \overset{*}{\longleftrightarrow}_{\{E, v \doteq w\}} t$. Conclude using (b).

(d) Prove that the sequent $E' \rightarrow s \doteq t$ is $G1^{nnf}_{=}$-provable iff $s \overset{*}{\longleftrightarrow}_E t$.

The above problem shows the completeness of the rewrite rule method for (universal) *equational logic*. This is an important current area of research. For more on this approach, see the article by Huet and Oppen, in Book, 1980, and Huet, 1980.

6.5 Craig's Interpolation Theorem

First, we define the concept of an interpolant.

6.5.1 Interpolants

If a formula of the form $A \supset B$ is valid, then it is not obvious that there is a formula I, called an *interpolant* of $A \supset B$, such that $A \supset I$ and $I \supset B$ are valid, and every predicate, constant, and free variable occurring in I also occurs both in A and B. As a matter of fact, the existence of an interpolant depends on the language. If we don't allow either the constant $\perp$ or equality ($\doteq$), the formula $(P \supset P) \supset (Q \supset Q)$ is valid, and yet, no formula I as above can be found. If we allow equality and make the exception that if $\doteq$ occurs in I, it does not necessarily occur in both A and B, then $I = \forall x(x \doteq x)$ does the job. Alternatively, $\neg \perp$ (true!) does the job.

Craig's interpolation theorem gives a full answer to the problem of the existence of an interpolant. An interesting application of Craig's interpolation theorem can be found in Oppen and Nelson's method for combining decision procedures. For details, see Nelson and Oppen, 1979, and Oppen, 1980b.

6.5.2 Craig's Interpolation Theorem Without Equality

First, we consider a first-order language without equality. For the next lemma, we assume that the constant $\perp$ (for false) is in the language, but that $\equiv$ is not. Let $LK_\perp$ be the extension of LK obtained by allowing the sequent $\perp\rightarrow$ as an axiom. It is easily checked that the cut elimination theorem also holds for $LK_\perp$ (see the problems).

During the proof of the key lemma, it will be necessary to replace in a proof all occurrences of a free variable y by a new variable z not occurring in the proof. This substitution process is defined as in definition 6.4.5, but for LK (and LK_e) rather than $G1^{nnf}$. Given a deduction tree T such that y does not occur bound in T, $T[z/y]$ is the result of replacing every sequent $\Gamma \rightarrow \Delta$ in T by $(\Gamma \rightarrow \Delta)[z/y]$. Similarly, $T[z/c]$ is the result of substituting a new variable z for all occurrences of a constant c in a proof T. The following technical lemma will be needed.

Lemma 6.5.1 Let $\Gamma \rightarrow \Delta$ be a sequent and T an LK-proof for $\Gamma \rightarrow \Delta$. Assume that y is a variable not occurring bound in the proof T. For any variable z not occurring in T, the result $T[z/y]$ of substituting z for all occurrences of y in T is a proof tree. Similarly, if c is a constant occurring in the proof tree T, $T[z/c]$ is a proof tree. The lemma also holds for LK_e-proofs.

Proof: Similar to that of lemma 6.4.3. □

The following lemma is the key to a constructive proof of the interpolation theorem.

Lemma 6.5.2 Let **L** be a first-order language without equality, without $\equiv$, and with constant $\perp$. Given any provable sequent $\Gamma \rightarrow \Delta$, let (Γ_1, Γ_2) and (Δ_1, Δ_2) be pairs of disjoint subsequences of Γ and Δ respectively, such that the union of Γ_1 and Γ_2 is Γ, and the union of Δ_1 and Δ_2 is Δ. Let us call $< \Gamma_1, \Delta_1, \Gamma_2, \Delta_2 >$ a *partition* of $\Gamma \to \Delta$. Then there is a formula C of $LK_\perp$ called an *interpolant* of $< \Gamma_1, \Delta_1, \Gamma_2, \Delta_2 >$ having the following properties:

(i) $\Gamma_1 \to \Delta_1, C$ and $C, \Gamma_2 \to \Delta_2$ are $LK_\perp$-provable.

(ii) All predicate symbols (except for $\perp$), constant symbols, and variables free in C occur both in $\Gamma_1 \cup \Delta_1$ and $\Gamma_2 \cup \Delta_2$.

Proof: We proceed by induction on cut-free proof trees. We treat some typical cases, leaving the others as an exercise.

(1) $\Gamma \to \Delta$ is an axiom. Hence, it is of the form $A \to A$. There are four cases. For $< \Gamma_1, \Delta_1, \Gamma_2, \Delta_2 >=< A, A, \emptyset, \emptyset >$, $C =\perp$; for $< \Gamma_1, \Delta_1, \Gamma_2, \Delta_2 >$ $=< \emptyset, \emptyset, A, A >$, $C = \neg \perp$; for $< \Gamma_1, \Delta_1, \Gamma_2, \Delta_2 >=< A, \emptyset, \emptyset, A >$, $C = A$; For $< \Gamma_1, \Delta_1, \Gamma_2, \Delta_2 >=< \emptyset, A, A, \emptyset >$, $C = \neg A$.

(2) The root inference is $\wedge : right$:

$$\frac{\Gamma \to \Delta, A \qquad \Gamma \to \Delta, B}{\Gamma \to \Delta, A \wedge B}$$

Assume the partition is

$$< \Gamma_1, (\Delta_1, A \wedge B), \Gamma_2, \Delta_2 >,$$

the case

$$< \Gamma_1, \Delta_1, \Gamma_2, (\Delta_2, A \wedge B) >$$

being similar. We have induced partitions $< \Gamma_1, (\Delta_1, A), \Gamma_2, \Delta_2 >$ and $< \Gamma_1, (\Delta_1, B), \Gamma_2, \Delta_2 >$. By the induction hypothesis, there are formulae C_1 and C_2 satisfying the conditions of the lemma. Since $\Gamma_1 \to \Delta_1, A, C_1$ and $\Gamma_1 \to \Delta_1, B, C_2$ are provable,

$$\Gamma_1 \to \Delta_1, A, C_1 \vee C_2$$

and

$$\Gamma_1 \to \Delta_1, B, C_1 \vee C_2$$

are provable (using $\vee : right$), and

$$\Gamma_1 \to \Delta_1, A \wedge B, C_1 \vee C_2$$

is provable, by $\wedge : right$. Since $C_1, \Gamma_2 \to \Delta_2$ and $C_2, \Gamma_2 \to \Delta_2$ are provable,

$$C_1 \vee C_2, \Gamma_2 \to \Delta_2$$

is provable by $\vee : left$. Then, we can take $C_1 \vee C_2$ as an interpolant.

(3) The root inference is $\forall : right$:

$$\frac{\Gamma \to \Delta, A[y/x]}{\Gamma \to \Delta, \forall x A}$$

where y does not occur free in the conclusion. Assume the partition is

$$< \Gamma_1, (\Delta_1, \forall x A), \Gamma_2, \Delta_2 >,$$

the case

$$< \Gamma_1, \Delta_1, \Gamma_2, (\Delta_2, \forall x A) >$$

being similar. By the induction hypothesis, there is an interpolant C such that $\Gamma_1 \to \Delta_1, A[y/x], C$ and $C, \Gamma_2 \to \Delta_2$ are provable. By condition (ii) of the lemma, C cannot contain y, since otherwise y would be in $\Gamma_1 \cup \Delta_1$, contradicting the fact that y does not occur free in $\Gamma \to \Delta, \forall x A$. Hence,

$$\Gamma_1 \to \Delta_1, \forall x A, C$$

is provable by $\forall : right$. Since by the induction hypothesis

$$C, \Gamma_2 \to \Delta_2$$

is provable, we can take C as an interpolant.

(4) The root inference is $\forall : left$:

$$\frac{A[t/x], \Gamma \to \Delta}{\forall x A, \Gamma \to \Delta}$$

where t is free for x in A.

This case is more complicated if t is not a variable. Indeed, one has to be careful to ensure condition (ii) of the lemma. Assume that the partition is $< (\forall x A, \Gamma_1), \Delta_1, \Gamma_2, \Delta_2 >$, the case $< \Gamma_1, \Delta_1, (\forall x A, \Gamma_2), \Delta_2 >$ being similar. By the induction hypothesis, there is a formula C such that

$$A[t/x], \Gamma_1 \to \Delta_1, C \quad \text{and} \quad C, \Gamma_2 \to \Delta_2$$

are provable. First, note that $\forall x A, \Gamma_1 \to \Delta_1, C$ is provable by $\forall : left$. If t is a variable z and z does not occur free in $\forall x A, \Gamma \to \Delta$, then by $\forall : right$, $\forall x A, \Gamma_1 \to \Delta_1, \forall z C$ is provable, and so is $\forall z C, \Gamma_2 \to \Delta_2$, by $\forall : left$. Since C contains predicate symbols, constants and free variables both occurring in $(A[z/x], \Gamma_1) \cup \Delta_1$ and $\Gamma_2 \cup \Delta_2$, and z does not occur free in $\forall x A, \Gamma \to \Delta$, the formula $\forall z C$ also satisfies condition (ii) of the lemma. If z occurs free in $(\forall x A, \Gamma_1) \cup \Delta_1$, then C can serve as an interpolant, since if it contains z, z is both in $(\forall x A, \Gamma_1) \cup \Delta_1$ and also in $\Gamma_2 \cup \Delta_2$ by the induction hypothesis.

If t is not a variable, we proceed as follows: Let $x_1, ...x_k$ and $c_1, ..., c_m$ be the variables and the constants in t which occur in C but do not occur in $(\forall xA, \Gamma_1) \cup \Delta_1$. In the proof of $A[t/x], \Gamma_1 \rightarrow \Delta_1, C$, replace all occurrences of $x_1, ...x_k$ and $c_1, ..., c_m$ in t and C by new variables $z_1, ..., z_{k+m}$ (not occurring in the proof). Let t' be the result of the substitution in the term t, and C' the result of the substitution in the formula C. By lemma 5.6.1, $A[t'/x], \Gamma_1 \rightarrow \Delta_1, C'$ is provable. But now, $z_1, ..., z_{k+m}$ do not occur free in $\forall xA, \Gamma_1 \rightarrow \Delta_1, \forall z_1...\forall z_{k+m}C'$, and so

$$\forall xA, \Gamma_1 \rightarrow \Delta_1, \forall z_1...\forall z_{k+m}C'$$

is provable (by applications of $\forall : right$ and $\forall : left$). But

$$\forall z_1...\forall z_{k+m}C', \Gamma_2 \rightarrow \Delta_2$$

is also provable (by applications of $\forall : left$). Hence,

$$\forall z_1...\forall z_{k+m}C'$$

can be used as an interpolant, since it also satisfies condition (ii) of the lemma. □

Note that the method used in lemma 6.5.2 does not guarantee that the function symbols occurring in C also occur both in $\Gamma_1 \cup \Delta_1$ and $\Gamma_2 \cup \Delta_2$.

We are now ready to prove Craig's interpolation theorem for LK.

Theorem 6.5.1 (Craig's interpolation theorem, without equality) Let **L** be a first-order language without equality and without $\equiv$. Let A and B be two **L**-formulae such that $A \supset B$ is LK-provable.

(a) If A and B contain some common predicate symbol, then there exists a formula C called an *interpolant* of $A \supset B$, such that all predicate, constant symbols, and free variables in C occur in both A and B, and both $A \supset C$ and $C \supset B$ are LK-provable.

(b) If A and B do not have any predicate symbol in common, either $A \rightarrow$ is LK-provable or $\rightarrow B$ is LK-provable.

Proof: Using the cut rule, it is obvious that $\rightarrow A \supset B$ is LK-provable iff $A \rightarrow B$ is. Consider the partition in which $\Gamma_1 = A$, $\Delta_1 = \emptyset$, $\Gamma_2 = \emptyset$ and $\Delta_2 = B$. By lemma 6.5.2, there is a formula C of $LK_\perp$ such that $A \rightarrow C$ and $C \rightarrow B$ are $LK_\perp$-provable, and all predicate, constant symbols, and free variables in C occur in A and B. If A and B have some predicate symbol P of rank n in common, let P' be the sentence

$$\forall y_1...\forall y_n(P(y_1, ..., y_n) \land \neg P(y_1, ...y_n)),$$

where $y_1, ..., y_n$ are new variables. Let C' be obtained by replacing all occurrences of $\perp$ in C by P'. Since P' is logically equivalent to $\perp$, it is not difficult to obtain LK-proofs of $A \rightarrow C'$ and of $C' \rightarrow B$. But then $\rightarrow A \supset C'$ and $\rightarrow C' \supset B$ are LK-provable, and the formula C' is the desired interpolant.

If A and B have no predicate symbols in common, the formula C given by lemma 6.5.2 only contains $\perp$ as an atom. It is easily shown by induction on the structure of C that either $\rightarrow C$ is $LK_\perp$-provable or $C \rightarrow$ is $LK_\perp$-provable. But then, using the cut rule, we can show that either $A \rightarrow$ is $LK_\perp$-provable, or $\rightarrow B$ is $LK_\perp$-provable. However, since neither A nor B contains $\perp$ and the cut elimination theorem holds for $LK_\perp$, a cut-free proof in $LK_\perp$ for either $A \rightarrow$ or $\rightarrow B$ is in fact an LK-proof. $\square$

EXAMPLE 6.5.1

Given the formula

$$(P(a) \land \forall x Q(x)) \supset (\forall y S(y) \lor Q(b)),$$

the formula $\forall x Q(x)$ is an interpolant.

6.5.3 Craig's Interpolation Theorem With Equality

We now consider Craig's interpolation theorem for first-order languages with equality. Because cuts cannot be completely eliminated in LK_e-proofs, the technique used in lemma 6.5.2 cannot be easily extended to the system LK_e. However, there is a way around, which is to show that a sequent $S = \Gamma \rightarrow \Delta$ is LK_e-provable if and only if the sequent $S_e, \Gamma \rightarrow \Delta$ is LK-provable, where S_e is a certain sequence of closed formulae called the *closed equality axioms* for $\Gamma \rightarrow \Delta$.

Definition 6.5.1 Given a sequent $S = \Gamma \rightarrow \Delta$, the set of *closed equality axioms for* S is the set S_e of closed formulae given below, for all predicate and function symbols occuring in S:

$$\forall x (x \doteq x)$$
$$\forall x_1 ... \forall x_n \forall y_1 ... \forall y_n (x_1 \doteq y_1 \land ... \land x_n \doteq y_n \supset f(x_1, ..., x_n) \doteq f(y_1, ..., y_n))$$
$$\forall x_1 ... \forall x_n \forall y_1 ... \forall y_n (x_1 \doteq y_1 \land ... \land x_n \doteq y_n \land P(x_1, ..., x_n) \supset P(y_1, ..., y_n))$$

Lemma 6.5.3 A sequent $S = \Gamma \rightarrow \Delta$ is LK_e-provable iff $S_e, \Gamma \rightarrow \Delta$ is LK-provable.

Proof: We sketch the proof, leaving the details as an exercise. First, we show that if S is LK_e-provable then $S_e, \Gamma \rightarrow \Delta$ is LK-provable. For this, we prove that for every equality axiom $\Gamma' \rightarrow \Delta'$ of LK_e used in the proof of S, the sequent $S_e, \Gamma' \rightarrow \Delta'$ is LK-provable. We conclude by induction on LK_e-proof trees.

Next, assume that $S_e, \Gamma \rightarrow \Delta$ is LK-provable. First, we show that every formula in S_e is LK_e-provable. We conclude by constructing an LK_e-proof of $\Gamma \rightarrow \Delta$ using cuts on formulae in S_e. $\square$

We can now generalize lemma 6.5.2 to LK_e as follows. Let $LK_{e,\perp}$ be the system obtained by allowing $\perp$ as a constant and the sequent $\perp \rightarrow$ as an axiom.

Lemma 6.5.4 Let **L** be a first-order language with equality and with constant $\perp$, but without $\equiv$. Given any LK_e-provable sequent $\Gamma \rightarrow \Delta$, let (Γ_1, Γ_2) and (Δ_1, Δ_2) be pairs of disjoint subsequences of Γ and Δ respectively, such that the union of Γ_1 and Γ_2 is Γ, and the union of Δ_1 and Δ_2 is Δ. $< \Gamma_1, \Delta_1, \Gamma_2, \Delta_2 >$ is called a *partition* of $\Gamma \rightarrow \Delta$. There is a formula C of $LK_{e,\perp}$ called an *interpolant* of $< \Gamma_1, \Delta_1, \Gamma_2, \Delta_2 >$ having the following properties:

(i) $\Gamma_1 \rightarrow \Delta_1, C$ and $C, \Gamma_2 \rightarrow \Delta_2$ are LK_e-provable.

(ii) All predicate symbols (except for $\doteq$), constant symbols, and variables free in C occur both in $\Gamma_1 \cup \Delta_1$ and $\Gamma_2 \cup \Delta_2$.

Proof: By lemma 6.5.3, $S = \Gamma \rightarrow \Delta$ is LK_e-provable iff $S_e, \Gamma \rightarrow \Delta$ is LK-provable. Let S_e^1 be the closed equality axioms corresponding to function and predicate symbols in $\Gamma_1 \cup \Delta_1$, and S_e^2 the closed equality axioms corresponding to function and predicate symbols in $\Gamma_2 \cup \Delta_2$. Clearly, $S_e = S_e^1 \cup S_e^2$. Consider the partition $< (S_e^1, \Gamma_1), \Delta_1, (S_e^2, \Gamma_2), \Delta_2 >$. By lemma 6.5.2, there is a formula C such that

$$S_e^1, \Gamma_1 \rightarrow \Delta_1, C \quad \text{and} \quad C, S_e^2, \Gamma_2 \rightarrow \Delta_2$$

are $LK_\perp$-provable, and the predicate symbols, constant symbols and free variables in C occur both in $S_e^1 \cup \Gamma_1 \cup \Delta_1$ and $S_e^2 \cup \Gamma_2 \cup \Delta_2$. By definition of S_e^1 and S_e^2, the predicate symbols (other than $\doteq$) in S_e^1 are those in $\Gamma_1 \cup \Delta_1$, and similarly for S_e^2 and $\Gamma_2 \cup \Delta_2$. Furthermore, all formulae in S_e^1 and S_e^2 being closed, there are no free variables occurring in them. Hence, the predicate symbols, constant and free variables in C occur both in $\Gamma_1 \cup \Delta_1$ and $\Gamma_2 \cup \Delta_2$. Using lemma 6.5.3 again, $\Gamma_1 \rightarrow \Delta_1, C$ and $C, \Gamma_2 \rightarrow \Delta_2$ are $LK_{e,\perp}$-provable. Now, we can replace all occurrence of $\perp$ in C by $\neg\forall z(z \doteq z)$, where z is a new variable, obtaining a formula C', and since $\perp$ is logically equivalent to $\neg\forall z(z \doteq z)$, it is easy to obtain LK_e-proofs for $\Gamma_1 \rightarrow \Delta_1, C'$ and $C', \Gamma_2 \rightarrow \Delta_2$. Taking C' as the interpolant this concludes the proof of the lemma. $\square$

Using lemma 6.5.4, Craig's interpolation theorem is generalized to languages with equality.

Theorem 6.5.2 (Craig's interpolation theorem, with equality) Let **L** be a first-order language with equality and without $\equiv$. Let A and B be two **L**-formulae such that $A \supset B$ is LK_e-provable. Then there exists a formula C called an *interpolant* of $A \supset B$, such that all predicate symbols except $\doteq$, constant symbols, and free variables in C occur in both A and B, and both $A \supset C$ and $C \supset B$ are LK_e-provable.

Proof: As in theorem 6.5.1, $A \supset B$ is LK_e-provable iff $A \to B$ is LK_e-provable. We apply lemma 6.5.4 to the partition in which $\Gamma_1 = A$, $\Delta_1 = \emptyset$, $\Gamma_2 = \emptyset$ and $\Delta_2 = B$. This time, because $\doteq$ is available, the result is obtained immediately from lemma 6.5.4. $\square$

Remark: As in theorem 6.5.2, our proof does not guarantee that the function symbols occurring in the interpolant also occur in both A and B. However, this can be achieved using a different proof technique presented in problem 5.6.7, which consists in replacing function symbols by predicate symbols. Hence, in case of a first-order language with equality, we obtain a stronger version of Craig's interpolation theorem, in which all predicate symbols (different from $\doteq$), function symbols, constant symbols, and free variables occurring in the interpolant C of $A \supset B$ occur in both A and B.

In the next section, we give two applications of Craig's interpolation theorem.

PROBLEMS

6.5.1. Finish the proof of lemma 6.5.1.

6.5.2. Finish the proof of lemma 6.5.2.

6.5.3. Give a proof for lemma 6.5.3.

6.5.4. Verify that the cut elimination theorem holds for $LK_{\perp}$, and for $LK_{e,\perp}$ (without essential cuts).

6.5.5. Provide the details in the proof of theorem 6.5.1 regarding the replacement of $\perp$ by $\forall y_1...\forall y_n(P(y_1,...,y_n) \land \neg P(y_1,...y_n))$.

6.5.6. Show that $\forall x \neg P(x) \lor Q(b)$ is an interpolant for

$$[(R \supset \exists x P(x)) \supset Q(b)] \supset [\forall x((S \land P(x)) \supset (S \land Q(b)))].$$

6.5.7. Using the proof technique presented in problem 5.6.7, which consists in replacing function symbols by predicate symbols, prove the stronger version of Craig's interpolation theorem, in which all predicate symbols (different from $\doteq$), function symbols, constant symbols, and free variables occurring in the interpolant C of $A \supset B$ occur in both A and B.

6.6 Beth's Definability Theorem

First, we consider what definability means.

6.6.1 Implicit and Explicit Definability

Let **L** be a first-order language with or without equality. Let $A_1, ..., A_m$ be closed formulae containing exactly the distinct predicate symbols $P_1, ..., P_k, Q$, where Q is not $\doteq$, but some of the P_i can be $\doteq$. Assume that Q has rank $n > 0$. We can view $A_1, ..., A_m$ as the axioms of a theory. The question of interest is whether Q is definable in terms of $P_1, ..., P_k$. First, we need to make precise what definable means. Let $A(P_1, ..., P_k, Q)$ be the conjunction $A_1 \wedge ... \wedge A_m$.

A first plausible criterion for definability is that, for any two **L**-structures **A** and **B** with the same domain M, and which assign the same interpretation to the predicate symbols $P_1, ..., P_k$,

$$\begin{gathered}\text{if} \quad \mathbf{A} \models A(P_1, ..., P_k, Q) \quad \text{and} \quad \mathbf{B} \models A(P_1, ..., P_k, Q), \\ \text{then for every } (a_1, ..., a_n) \in M^n, \\ \mathbf{A} \models Q(a_1, ..., a_n) \quad \text{iff} \quad \mathbf{B} \models Q(a_1, ..., a_n).\end{gathered}$$

The idea is that given any two interpretations of the predicate symbols Q, if these two interpretations make $A(P_1, ..., P_k, Q)$ true then they must agree on Q. This is what is called *implicit definability*.

A seemingly stronger criterion for definability is the following:

Definition 6.6.1 Assume that there is an **L**-formula $D(P_1, ..., P_k)$ whose set of free variables is a subset of $\{x_1, ..., x_n\}$ and whose predicate symbols are among $P_1, ..., P_k$, and which contains at least one of the P_i. We say that Q is *defined explicitly* from $P_1, ..., P_k$ in the theory based on $A_1, ..., A_m$ iff the following is provable (or equivalently valid, by the completeness theorem):

$$\rightarrow A(P_1, ..., P_k, Q) \supset \forall x_1 ... \forall x_n (Q(x_1, ..., x_n) \equiv D(P_1, ..., P_k)),$$

where $A(P_1, ..., P_k, Q)$ is the conjunction $A_1 \wedge ... \wedge A_m$.

We can modify the definition of implicit definability so that it refers to a single structure as opposed to a pair of structures, by using a new copy Q' of the predicate symbol Q, explained as follows.

Definition 6.6.2 Let $P_1, ..., P_k, Q, Q'$ be distinct predicate symbols, and let $A(P_1, ..., P_k, Q')$ be the result of substituting Q' for Q in $A(P_1, ..., P_k, Q)$. We say that Q is *defined implicitly* from $P_1, ..., P_k$ in the theory based on $A_1, ..., A_m$ iff the following formula is valid (or equivalently provable):

$$A(P_1, ..., P_k, Q) \wedge A(P_1, ..., P_k, Q') \supset \forall x_1 ... \forall x_n (Q(x_1, ..., x_n) \equiv Q'(x_1, ..., x_n)),$$

where $A(P_1, ..., P_k, Q)$ is the conjunction $A_1 \wedge ... \wedge A_m$.

EXAMPLE 6.6.1

Let $\mathbf{L}$ be the language with equality having 0 as a constant, S as a unary function symbol, $+$ as a binary function symbol, and $<$ as a binary predicate symbol. Let $A(\doteq, <)$ be the conjunction of the following closed formulae:

$$\forall x \neg (S(x) \doteq 0)$$
$$\forall x \forall y (S(x) \doteq S(y) \supset x \doteq y)$$
$$\forall x (x + 0 \doteq x)$$
$$\forall x \forall y (x + S(y) \doteq S(x + y))$$
$$\forall x (0 < Sx)$$

The predicate symbol $<$ is not definable implicitly from $\doteq$ and $A(\doteq, <)$. Indeed, we can define two structures $\mathbf{A}$ and $\mathbf{B}$ with domain $\mathbf{N}$, in which 0, S and $+$ receive the same natural interpretation, but in the first structure, $<$ is interpreted as the strict order on $\mathbf{N}$, whereas in the second, we interpret $<$ as the predicate such that for all $x, y \in \mathbf{N}$, $x < y$ iff $x = 0$. Both $\mathbf{A}$ and $\mathbf{B}$ are models of $A(\doteq, <)$, but $<$ is interpreted differently.

On the other hand, if we add to $A(\doteq, <)$ the sentence

$$\forall x \forall y (x < y \equiv \exists z (y \doteq x + S(z))),$$

then $D = \exists z (y \doteq x + S(z))$ defines $<$ explicitly.

6.6.2 Explicit Definability Implies Implicit Definability

It is natural to ask whether the concepts of implicit and explicit definability are related. First, it is not difficult to see that explicit definability implies implicit definability. Indeed, we show that if

$$A(P_1, ..., P_k, Q) \supset \forall x_1 ... \forall x_n (Q(x_1, ..., x_n) \equiv D(P_1, ..., P_k))$$

is valid, then $A(P_1, ..., P_k, Q)$ defines Q implicitly. For if $A(P_1, ..., P_k, Q) \wedge A(P_1, ..., P_k, Q')$ is valid in any model $\mathbf{A}$, then

$$\forall x_1 ... \forall x_n (Q(x_1, ..., x_n) \equiv D(P_1, ..., P_k)) \quad \text{and}$$
$$\forall x_1 ... \forall x_n (Q'(x_1, ..., x_n) \equiv D(P_1, ..., P_k))$$

are valid in $\mathbf{A}$, and this implies that

$$\forall x_1 ... \forall x_n (Q(x_1, ..., x_n) \equiv Q'(x_1, ..., x_n))$$

is valid in **A**.

The fact that explicit definability implies implicit definability yields a method known as *Padoa's method*, to show that a predicate Q is not definable explicitly from $P_1, ..., P_k$ in the theory based on $A_1, ..., A_m$:

Find two structures with the same domain and assigning the same interpretation to the predicate symbols $P_1, ..., P_k$, but assigning different interpretations to Q.

However, it is not as obvious that implicit definability implies explicit definability. But this is the case, as shown by Beth's theorem. The proof given below that uses Craig's interpolation theorem even gives the defining formula $D(P_1, ..., P_k)$ constructively.

6.6.3 Beth's Definability Theorem, Without Equality

First, we consider the case of a first-order language without equality.

Theorem 6.6.1 (Beth's definability theorem, without equality) Let **L** be a first-order language without equality. Let $A(P_1, ..., P_k, Q)$ be a closed formula containing predicate symbols among the distinct predicate symbols $P_1, ..., P_k$, Q, where Q has rank $n > 0$. Assume that Q is defined implicitly from $P_1, ..., P_k$ by the sentence $A(P_1, ..., P_k, Q)$.

(a) If one of the predicate symbols P_i actually occurs in $A(P_1, ..., P_k, Q)$, then there is a formula $D(P_1, ..., P_k)$ defining Q explicitly from $P_1, ..., P_k$.

(b) If none of the predicate symbols $P_1, ..., P_k$ occur in $A(P_1, ..., P_k)$, then either

$$\models A(P_1, ..., P_k, Q) \supset \forall x_1 ... \forall x_n Q(x_1, ..., x_n) \quad \text{or}$$
$$\models A(P_1, ..., P_k, Q) \supset \forall x_1 ... \forall x_n \neg Q(x_1, ..., x_n).$$

Proof: Assume that

$$A(P_1, ..., P_k, Q) \land A(P_1, ..., P_k, Q') \supset \forall x_1 ... \forall x_n (Q(x_1, ..., x_n) \equiv Q'(x_1, ..., x_n))$$

is valid. Since $A(P_1, ..., P_k, Q)$ and $A(P_1, ..., P_k, Q')$ are closed,

$$A(P_1, ..., P_k, Q) \land A(P_1, ..., P_k, Q') \supset (Q(x_1, ..., x_n) \equiv Q'(x_1, ..., x_n))$$

is also valid. This formula is of the form $(A \land B) \supset (C \equiv D)$. It is an easy exercise to show that if $(A \land B) \supset (C \equiv D)$ is valid, then $(A \land C) \supset (B \supset D)$ is also valid. Hence,

$$(1) \quad (A(P_1, ..., P_k, Q) \land Q(x_1, ..., x_n)) \supset (A(P_1, ..., P_k, Q') \supset Q'(x_1, ..., x_n))$$

is valid.

(a) If some of the predicate symbols P_i occurs in $A(P_1, ..., P_k, Q)$, by Craig's theorem (theorem 6.5.1) applied to (1), there is a formula C containing only predicate symbols, constant symbols, and free variables occurring in both $A(P_1, ..., P_k, Q) \wedge Q(x_1, ..., x_n)$ and $A(P_1, ..., P_k, Q') \supset Q'(x_1, ..., x_n)$, and such that the following are valid:

$$(A(P_1, ..., P_k, Q) \wedge Q(x_1, ..., x_n)) \supset C, \tag{2}$$

and

$$C \supset (A(P_1, ..., P_k, Q') \supset Q'(x_1, ..., x_n)). \tag{3}$$

Since $A(P_1, ..., P_k, Q) \wedge Q(x_1, ..., x_n)$ does not contain Q' and $A(P_1, ..., P_k, Q')$ $\supset Q'(x_1, ..., x_n)$ does not contain Q, the formula C only contains predicate symbols among $P_1, ..., P_k$ and free variables among $x_1, ..., x_n$.

By substituting Q for Q' in (3), we also have the valid formula

$$C \supset (A(P_1, ..., P_k, Q) \supset Q(x_1, ..., x_n)), \tag{4}$$

which implies the valid formula

$$A(P_1, ..., P_k, Q) \supset (C \supset Q(x_1, ..., x_n)). \tag{5}$$

But (2) implies the validity of

$$A(P_1, ..., P_k, Q) \supset (Q(x_1, ..., x_n) \supset C). \tag{6}$$

The validity of (5) and (6) implies that

$$A(P_1, ..., P_k, Q) \supset (C \equiv Q(x_1, ..., x_n))$$

is valid, which in turns implies that C defines Q explicitly from $P_1, ..., P_k$.

(b) If none of $P_1, ..., P_k$ occurs in $A(P_1, ..., P_k, Q)$, then by Craig's theorem (theorem 6.5.1), part (ii), either

$$\neg(A(P_1, ..., P_k, Q) \wedge Q(x_1, ..., x_n)) \tag{7}$$

is valid, or

$$A(P_1, ..., P_k, Q') \supset Q'(x_1, ..., x_n) \tag{8}$$

is valid.

Using propositional logic, either

$$A(P_1, ..., P_k, Q) \supset \neg Q(x_1, ..., x_n)$$

is valid, or

$$A(P_1, ..., P_k, Q) \supset Q(x_1, ..., x_n)$$

is valid, which implies part (b) of the theorem. □

6.6.4 Beth's Definability Theorem, With Equality

We now consider Beth's definability theorem for a first-order language with equality. This time, we can define either a predicate symbol, or a function symbol, or a constant.

Theorem 6.6.2 (Beth's definability theorem, with equality) Let **L** be a first-order language with equality.

(a) Let $A(P_1, ..., P_k, Q)$ be a closed formula possibly containing equality and containing predicate symbols among the distinct predicate symbols $P_1, ..., P_k, Q$ (different from $\doteq$), where Q has rank $n > 0$. Assume that Q is defined implicitly from $P_1, ..., P_k$ by the sentence $A(P_1, ..., P_k, Q)$. Then there is a formula $D(P_1, ..., P_k)$ defining Q explicitly from $P_1, ..., P_k$.

(b) Let $A(P_1, ..., P_k, f)$ be a closed formula possibly containing equality, and containing predicate symbols among the distinct predicate symbols $P_1, ..., P_k$ (different from $\doteq$), and containing the function or constant symbol f of rank $n \geq 0$. Assume that f is defined implicitly from $P_1, ..., P_k$ by the sentence $A(P_1, ..., P_k, f)$, which means that the following formula is valid, where f' is a new copy of f:

$$A(P_1, ..., P_k, f) \wedge A(P_1, ..., P_k, f') \supset \forall x_1 ... \forall x_n (f(x_1, ..., x_n) \doteq f'(x_1, ..., x_n)).$$

Then there is a formula $D(P_1, ..., P_k)$ whose set of free variables is among $x_1, ..., x_n$ and not containing f (or f') defining f explicitly, in the sense that the following formula is valid:

$$A(P_1, ..., P_k, f) \supset \forall x_1 ... \forall x_n \forall y ((f(x_1, ..., x_n) \doteq y) \equiv D(P_1, ..., P_k)).$$

Proof: The proof of (a) is similar to that of theorem 6.6.1(a), but using theorem 6.5.2, which yields $D(P_1, ..., P_k)$ in all cases.

To prove (b), observe that $f(x_1, ..., x_n) \doteq f'(x_1, ..., x_n)$ is equivalent to

$$\forall y ((f(x_1, ..., x_n) \doteq y) \equiv (f'(x_1, ..., x_n) \doteq y)).$$

We conclude by applying the reasoning used in part (a) with $f(x_1, ..., x_n) \doteq y$ instead of $Q(x_1, ..., x_n)$. □

The last application of Craig's interpolation theorem presented in the next section is Robinson's joint consistency theorem.

PROBLEMS

6.6.1. Show that in definition 6.6.2, the definability condition can be relaxed to

$$A(P_1, ..., P_k, Q) \land A(P_1, ..., P_k, Q') \supset \forall x_1 ... \forall x_n (Q(x_1, ..., x_n) \supset Q'(x_1, ..., x_n)).$$

6.6.2. Give the details of the proof that explicit definability implies implicit definability.

6.7 Robinson's Joint Consistency Theorem

Let **L** be a first-order language with or without equality, and let $\mathbf{L}_1$ and $\mathbf{L}_2$ be two expansions of **L** such that $\mathbf{L} = \mathbf{L}_1 \cap \mathbf{L}_2$. Also, let S_1 be a set of $\mathbf{L}_1$-sentences, S_2 a set of $\mathbf{L}_2$-sentences, and let $S = S_1 \cap S_2$. If S_1 and S_2 are both consistent, their union $S_1 \cup S_2$ is not necessarily consistent. Indeed, if S is *incomplete*, that is, there is some **L**-sentence C such that neither $S \to C$ nor $S \to \neg C$ is provable, S_1 could contain C and S_2 could contain $\neg C$, and $S_1 \cup S_2$ would be inconsistent. A concrete illustration of this phenomenon can be given using Gödel's incompleteness theorem for Peano's arithmetic, which states that there is a sentence C of the language of arithmetic such that neither $A_P \to C$ nor $A_P \to \neg C$ is provable, where A_P consists of the axioms of Peano's arithmetic (see example 5.6.3). (For a treatment of Gödel's incompleteness theorems, see Enderton, 1972; Kleene, 1952; Shoenfield, 1967; or Monk, 1976.) Since Peano's arithmetic is incomplete, then $\{A_P, C\}$ and $\{A_P, \neg C\}$ are both consistent, but their union is inconsistent.

A. Robinson's theorem shows that inconsistency does not arise if S is complete. Actually, one has to be a little careful about the presence in the language of function symbols or of the equality predicate $\doteq$.

Theorem 6.7.1 (Robinson's joint consistency theorem) Let **L** be a first-order language either without function symbols and without equality, or with equality (and possibly function symbols). Let $\mathbf{L}_1$ and $\mathbf{L}_2$ be two expansions of **L** such that $\mathbf{L} = \mathbf{L}_1 \cap \mathbf{L}_2$. Let S_1 be a set of $\mathbf{L}_1$-sentences, S_2 a set of $\mathbf{L}_2$-sentences, and let $S = S_1 \cap S_2$. If S_1 and S_2 are consistent and S is complete, that is, for every closed **L**-formula C, either $\vdash S \to C$, or $\vdash S \to \neg C$, then the union $S_1 \cup S_2$ of S_1 and S_2 is consistent.

Proof: Assume that $S_1 \cup S_2$ is inconsistent. Then $\models S_1 \cup S_2 \to$. By the completeness theorem (theorem 5.6.1), there is a finite subsequent of $S_1 \cup S_2 \to$ that is provable. Let $A_1, ..., A_m, B_1, ..., B_n \to$ be such a sequent, where $A_1, ..., A_m \in S_1$, and $B_1, ..., B_n \in S_2$. It is immediate that

$$\vdash (A_1 \land ... \land A_m) \supset \neg(B_1 \land ... \land B_n).$$

We apply Craig's interpolation theorem (theorem 6.5.1) to this formula.

First, we consider the case where **L** does not contain function symbols and does not contain equality. Then, if $A_1 \wedge ... \wedge A_m$ and $\neg(B_1 \wedge ... \wedge B_n)$ do not have any predicate symbol in common, either

$$\vdash \neg(A_1 \wedge ... \wedge A_m), \text{ or}$$
$$\vdash \neg(B_1 \wedge ... \wedge B_n).$$

In the first case, the consistency of S_1 is contradicted, in the second case, the consistency if S_2 is contradicted.

If $(A_1 \wedge ... \wedge A_m)$ and $\neg(B_1 \wedge ... \wedge B_n)$ have some predicate in common, then there is a formula C such that

$$(1) \qquad \vdash (A_1 \wedge ... \wedge A_m) \supset C,$$

and

$$(2) \qquad \vdash C \supset \neg(B_1 \wedge ... \wedge B_n),$$

and the predicate symbols, constant symbols and variables free in C are both in $A_1 \wedge ... \wedge A_m$ and $\neg(B_1 \wedge ... \wedge B_n)$. Since these formulae are closed, C is also a closed formula, and since $\mathbf{L} = \mathbf{L}_1 \cap \mathbf{L}_2$, C is a closed **L**-formula. Since S is complete, either $\vdash S \rightarrow C$ or $\vdash S \rightarrow \neg C$. In the first case, by (2)

$$\vdash S \rightarrow \neg(B_1 \wedge ... \wedge B_n),$$

contradicting the consistency of S_2. In the second case, by (1)

$$\vdash S \rightarrow \neg(A_1 \wedge ... \wedge A_m),$$

contradicting the consistency of S_1.

If **L** is a language with equality, we need the strong form of Craig's interpolation theorem mentioned as a remark after theorem 6.5.2, which states that the all predicate, function and constant symbols occurring in an interpolant C of $A \supset B$ occur in both A and B. The rest of the proof is as above. $\square$

Another slightly more general version of Robinson's joint consistency theorem is given in problem 6.7.1.

PROBLEMS

6.7.1. Prove the following version of Robinson's joint consistency theorem:

Let $\mathbf{L}$ be a first-order language either without function symbols and without equality, or with equality (and possibly function symbols). Let $\mathbf{L}_1$ and $\mathbf{L}_2$ be two expansions of $\mathbf{L}$ such that $\mathbf{L} = \mathbf{L}_1 \cap \mathbf{L}_2$. Let S_1 be a set of $\mathbf{L}_1$-sentences, and let S_2 a set of $\mathbf{L}_2$-sentences. Assume that S_1 and S_2 are consistent. Then the union $S_1 \cup S_2$ of S_1 and S_2 is consistent iff for every closed $\mathbf{L}$-formula C, either $S_1 \rightarrow C$ is not provable, or $S_2 \rightarrow \neg C$ is not provable.

6.7.2. Prove that the version of Robinson's joint consistency theorem given in problem 6.7.1 implies Craig's interpolation theorem.

Hint: Let $A \supset B$ be a provable formula. Let $S_1 = \{C \mid \ \vdash A \rightarrow C\}$, and $S_2 = \{C \mid \ \vdash \neg B \rightarrow C\}$. Then $S_1 \cup S_2$ is inconsistent.

Notes and Suggestions for Further Reading

Gentzen's cut elimination theorem is one of the jewels of proof theory. Originally, Gentzen's motivation was to provide constructive consistency proofs, and the cut elimination theorem is one of the main tools.

Gentzen's original proof can be found in Szabo, 1969, and other proofs are in Kleene, 1952, and Takeuti, 1975. A very elegant proof can also be found in Smullyan, 1968. The proof given in Section 6.4 is inspired by Schwichtenberg and Tait (Barwise, 1977, Tait, 1968).

Craig himself used Gentzen systems for proving his interpolation theorem. We have followed a method due to Maehara sketched in Takeuti, 1975, similar to the method used in Kleene, 1967. There are model-theoretic proofs of Craig's theorem, Beth's definability theorem, and Robinson's joint consistency theorem. The reader is referred to Chang and Keisler, 1973, or Shoenfield, 1967.

The reader interested in proof theory should also read the article by Schwichtenberg in Barwise, 1977. For an interesting analysis of analytic versus nonanalytic proofs, the reader is referred to the article by Frank Pfenning, in Shostak, 1984a.