On the Lengths of Proofs in the Propositional Calculus

Preliminary Version

Stephen Cook and Robert Reckhow
University of Toronto

I. Introduction

One of the most important open questions in the field of computational complexity is the question of whether there is a polynomial time decision procedure for the classical propositional calculus. The importance stems from theorem 1 in Cook [1971a], which demonstrates the equivalence of this question to the question of whether $P = NP$, and from results in Cook [1971a] & Karp [1972] which show that an answer either way to the question $P = NP$ would have strong implications concerning the complexity of many combinatorial problems of interest in Computer Science.

The purpose of the present paper is to study a question related to the complexity of decision procedures for the propositional calculus; namely, the complexity of proof systems for the propositional calculus. The fundamental issue here is whether there exists any proof system, and a polynomial $p(n)$ such that every valid formula has a proof of length not exceeding $p(n)$, where n is the length of the formula. Theorem 1 below helps establish the importance of this question. For the purposes of this theorem, we give the following definitions.

A proof system is a function F from the set Σ^* of strings on some finite alphabet Σ onto the set of valid propositional formulas such that F can be computed in polynomial time by a Turing machine. If $F(w) = A$, then w is said to be a proof of A in the system.

All ordinary proof systems for tautologies can easily be made to fit this definition by regarding the proofs in the system as strings on some alphabet, and the function F would take a proof into the formula proved. If a string w did not code a proof, then define $F(w) = p \vee \neg p$. If the system is a refutation system for inconsistent formulas, then one can regard a refutation of $\neg A$ as a proof of A.

Proposition: The set of tautologies is in NP if and only if there exists a super proof system.

The proof is almost immediate. If there exists a super proof system F, then the nondeterministic polynomial time proof procedure for the tautologies would consist in guessing at the proof w for an input formula A, and then checking that $F(w) = A$. Conversely, if the tautologies are in NP, then a super proof system can be obtained from a nondeterministic polynomial time procedure for the tautologies by letting every accepting computation of an input formula A be a proof of A.

Theorem 1 The class NP is closed under complements if and only if there exists a super proof system for the tautologies.

Proof: Suppose NP is closed under complements. Since the set of falsifiable formulas is in NP, it follows that the set of tautologies is in NP, and by the above lemma, there exists a super proof system.

Conversely, suppose there exists a super proof system. Then the tautologies are in NP. Consider an arbitrary set L in NP, and let Z be a single tape nondeterministic Turing machine which accepts L in polynomial time. Given an input string w to Z, let $A(w)$ be the propositional formula constructed in the proof of theorem 1 in Cook [1971a]. Then $A(w)$ is satisfiable iff Z accepts w. Hence $\neg A(w)$ is a tautology iff $w \in L^c$, where L^c is the complement of L. Since there is a nondeterministic procedure to accept the tautologies, and since $A(w)$ can be constructed in polynomial time, it follows that L^c is in NP.

Thus there are two main reasons for studying the complexity of proof systems. If we succeed in finding a super proof system, then it would follow that NP is closed under complements, which would have very interesting implications for each of the combinatorial problems discussed in

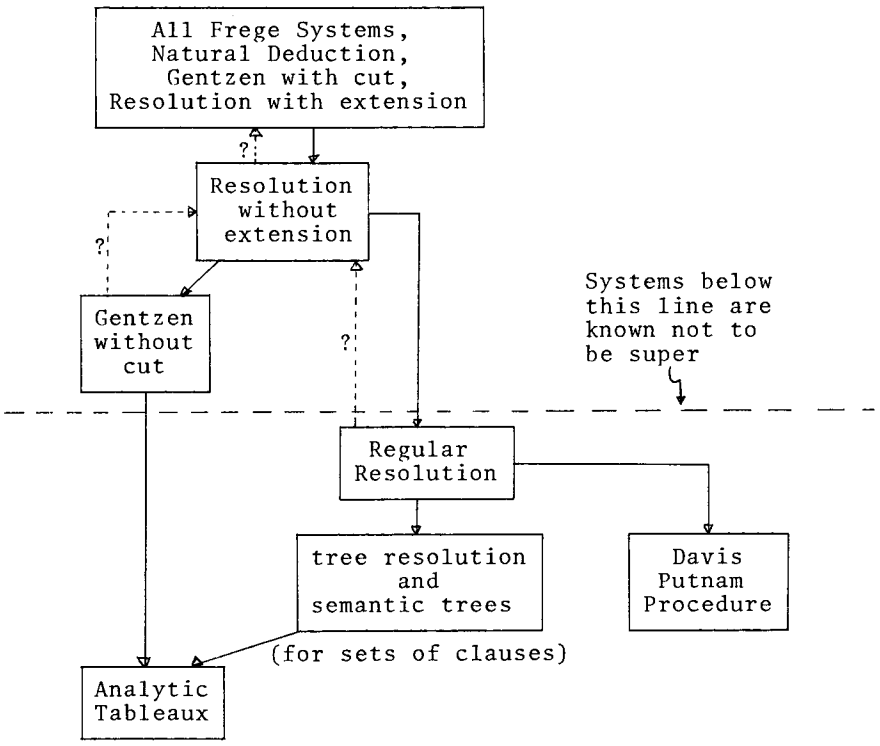


Figure 1 The Relative Strengths of Various Proof Systems

The strongest proof system (i.e. those with shortest proofs) are at the top. An arrow $[S_1] \rightarrow [S_2]$ indicates that system S_1 can simulate system S_2, in the sense that for some polynomial p(n) (never more than fourth degree), for every proof or refutation D in S_1 of a formula A there is a proof or refutation D' in S_2 of a suitable translation of A such that $|D'| \leq p(|D|)$. A dashed line $\overset{?}{\dashrightarrow}$ indicates it is not known whether there is a reverse polynomial simulation. Where no such line is drawn, it is known there is no reverse polynomial simulation. Systems appearing in the same box can simulate each other in this sense. Systems below the horizontal dashed line are known not to be super (i.e. there is no polynomial bound on the length of the shortest proof of A as a function of the length of A).

Karp [1972]. (For example, it would show the existence of a proof system, such as a list of invariants, which would provide a short proof that any two non-isomorphic graphs are in fact not isomorphic.) If, on the other hand, we could show that no super proof system exists, we would then have a proof that $P \neq NP$.

In the remainder of the paper we will partially classify according to relative strength most of the major proof systems that have been proposed for the propositional calculus. We will prove (or report from other papers) results of two kinds. First, we will present lower bounds on the minimum proof lengths for some systems. These will show that some systems, such as Smullyan's analytic tableaux and the Davis-Putnam decision procedure (regarded as a proof system) are definitely not super. Second, we will present polynomial simulation results between pairs of proof systems.

Figure 1 presents a summary of our results. The unfamiliar terms in the figure will be explained later. In particular, a Frege System is a general kind of proof system which includes many of the standard systems appearing in logic textbooks, such as Kleene [1952].

The major result which emerges from this paper is that most major proof systems (namely those indicated in the top box in figure 1) are equivalent, in the sense that given any two systems there is a polynomial p(n) so that for any proof or refutation D of a formula A in the first system there is a proof or refutation D' of a suitable translation of A in the second system such that $|D'| \leq p(|D|)$. (Translation of A is only necessary if the systems use different logical connectives,

or one of the systems is resolution.) In particular, any one system in the top box is super if and only if all are super. Furthermore, every proof system we have come across in the literature specifically for proving validity or unsatisfiability of propositional formulas (or systems for predicate formulas, suitably restricted) can be simulated by systems in this equivalence class, with at worst a polynomial increase in proof length. Of course powerful formal theories, such as Zermelo-Fraenkel set theory, can be used as proof systems for tautologies, and it seems reasonable to guess that these have no polynomial simulation by members of the equivalence class.

Nevertheless, the methods incorporated in the equivalence class at the top of Figure 1 are stable and powerful. None of the specific sets of formulas defined in this paper or in the papers referred to here will show these systems are not super. In fact a proof that these systems are not super would be a very interesting and probably a deep result.

We propose studying these systems, both from the point of view of trying to find examples with no short proofs, and by trying to find more invariant characterizations of the proof methods embodied in these systems. Another interesting research problem is to study proof systems for some of the complements of the combinatorial problems in *NP* listed in Karp [1972], and try to find natural proof systems for these problems which are "equivalent", in an appropriate sense, to the powerful systems for the propositional calculus.

We remark that to us the most interesting simulation result is that the Frege systems, such as the standard system in Kleene [1952], can simulate natural deduction or Gentzen with cut, with only a linear or quadratic increase in proof length. The surprise comes because natural deduction incorporates the deduction theorem as a rule of inference, and the standard method of proving the deduction theorem for a Frege system seems to involve an exponential increase in proof length.

II. Languages for Propositional Calculus

Consider the language L defined by the following grammar. The alphabet for L is $\{P,0,1,(,),T,F,\neg,\vee,\subset,\supset,\equiv,\&,|,\not\equiv,\not\supset,\not\subset,\downarrow\}$.

<atom> → P, <atom>0, <atom>1 (note: A → B,C abbreviates A → B and A → C)
<nullary connective> → T,F
<unary connective> → ¬
<v-like connective> → ∨,⊂,⊃,&,|,⊅,⊄,↓
<≡-like connective> → ≡,≢
<binary connective> → <v-like connective>, <≡-like connective>
<connective> → <nullary connective>, <unary connective>, <binary connective>
<atomic formula> → <atom>, <nullary connective>
<formula> → <atomic formula>, <unary connective><formula>, (<formula> <binary connective><formula>)

If κ is a set of connectives, then L_κ is the restriction of L to the alphabet $\{P,0,1,(,)\} \cup \kappa$. In what follows, lower case letters p,q,r, etc. will represent atoms, and upper case letters A,B,C,A_1,A_2, etc. will represent formulas.

If A is a non-atomic formula, then A has one of the forms $\neg B$ or $(B \circ C)$, where B and C are formulas and $\circ$ is a binary connective. The <u>principal subformulas</u> of A are B if A is $\neg B$, or B and C if A is $(B \circ C)$. The <u>principal connective</u> of A is $\neg$ if A is $\neg B$, or $\circ$ if A is $(B \circ C)$. The set $sub(A)$ of <u>subformulas</u> of A is defined inductively by:

$$sub(A) = \{A\} \cup \begin{cases} \phi \text{ (the empty set) if } A \text{ is an atomic formula} \\ sub(B) \text{ if } A \text{ is } \neg B \\ sub(B) \cup sub(C) \text{ if } A \text{ is } (B \circ C) \end{cases}$$

If S is a set of formulas, then $sub(S) = \bigcup_{A \in S} sub(A)$. The set $at(S)$ of atoms of S is $sub(S) \cap (P\{0,1\}^*)$, the set of subformulas of S which are atoms.

If $A_1,\ldots,A_k$ are formulas and $p_1,\ldots,p_k$ are distinct atoms, then the <u>substitution</u> $\sigma = \frac{A_1,\ldots,A_k}{p_1,\ldots,p_k}$ is the mapping from formulas to formulas such that $\sigma(B)$ (usually written $B\sigma$) is the formula obtained by (simultaneously) replacing all occurrences of each p_i in B by the corresponding A_i. (The result of this substitution is necessarily a formula, since a formula can appear anywhere that an atom can appear in a formula.) The formula $B\sigma$ is said to be an <u>instance</u> of B. If S is a set of formulas, then $S\sigma$ is the set of formulas obtained by applying σ to each formula in S. If $A_1,\ldots,A_k$ are in formulas in L_κ, then $\sigma = \frac{A_1,\ldots,A_k}{p_1,\ldots,p_k}$ is a substitution in L_κ. If $\sigma = \frac{q_1,\ldots,q_k}{p_1,\ldots,p_k}$, $q_1,\ldots,q_k$ are distinct atoms, and $\{q_1,\ldots,q_k\} \cap (at(S)-\{p_1,\ldots,p_k\}) = \phi$ (i.e. no q_i which is not a p_j is an atom of S), then σ is called a <u>renaming</u> (for S).

If A is a set of atoms, then a truth assignment to A is a mapping $\tau : A \to \{t,f\}$ from A to the set of truth values: t (ture) and f (false). If S is a set of formulas and τ is a truth assignment to $at(S)$, then τ can be extended to $sub(S)$ according to the following inductive definition:

$$\tau(T) = t, \quad \tau(F) = f$$

$$\tau(\neg B) = \begin{cases} f & \text{if } \tau(B) = t \\ t & \text{if } \tau(B) = f \end{cases}$$

If A is $(B \circ C)$ then $\tau(A)$ is given by this table

$\tau(B)$	$\tau(C)$	$\tau(A)$										
		$\circ$:	$\vee$	$\subset$	$\supset$	$\equiv$	$\&$		$\not\equiv$	$\not\supset$	$\not\subset$	$\downarrow$
t	t		t	t	t	t	t	f	f	f	f	f
t	f		t	t	f	f	f	t	t	t	f	f
f	t		t	f	t	f	f	t	t	f	t	f
f	f		f	t	t	t	f	t	f	f	f	t

We note that the ten binary connectives listed represent those ten of the sixteen binary truth functions which depend on both arguments. The six omitted are the two projection functions, the two negated projection functions, and the two constant functions. These six can be represented using unary and nullary connectives.

Truth assignment τ to $at(A)$ satisfies (falsifies) A iff $\tau(A) = t(f)$. Formula A is satisfiable (falsifiable) iff there is a truth assignment which satisfies (falsifies) A. Formula A is unsatisfiable, also inconsistent (valid, also a tautology) iff A is not satisfiable (falsifiable). Note that A is a valid (inconsistent) iff $\neg A$ is inconsistent (valid). A formula which is both satisfiable and falsifiable is said to be contingent.

Set of formulas S logically implies formula A (denoted $S \models A$) iff every truth assignment to $at(S \cup \{A\})$ which does not falsify any formula in S, satisfies A. (Note that this definition says that for $S = \phi$, $\phi \models A$ iff A is a tautology. $\phi \models A$ is abbreviated to $\models A$.) Formulas A and B are logically equivalent (denoted $A \sim B$) iff $A \models B$ and $B \models A$.

Let A be a set of atoms where $|A| = n$. Let $\langle p_1,\ldots,p_n\rangle$ be the atoms of A, ordered lexicographically. There is a 1-to-1 correspondence between 2^n distinct truth assignments to A and the 2^n distinct n-tuples of t and f, given by $\tau \leftrightarrow \langle\tau(p_1),\ldots,\tau(p_n)\rangle$. A truth function of n variables is a function $\theta : \{t,f\}^n \to \{t,f\}$ from n-tuples of t and f (equivalently, truth assignments to a set of n atoms) to $\{t,f\}$. There are 2^{2^n} distinct truth functions of n variables. If $|at(A)| = n$, then the truth function expressed by A is the truth function θ_A of n variables defined by: $\theta_A(\tau) = \tau(A)$. If $A\sigma$ is a renaming of A, then $\theta_{A\sigma}$ comes from θ_A by some permutation of the arguments. If $\sigma = \frac{A_1,\ldots,A_n}{p_1,\ldots,p_n}$, where $p_1,\ldots,p_n$ are the atoms of B in lexicographic order then $\theta_{B\sigma}(\vec{x}) = \theta_B(\theta_{A_1}(\vec{x}),\ldots,\theta_{A_n}(\vec{x}))$.

Let κ be a set of connectives. κ is adequate iff for every truth function θ there is a formula A in L_κ which expresses θ. κ is minimally adequate iff κ is adequate and no proper subset of κ is adequate. It can be shown that there are 26 minimally adequate sets of connectives $(\{|\},\{+\},\{\neg,\vee\},\{\neg,\subset\},\{\neg,\supset\},\{\neg,\&\},\{\neg,\not\equiv\},\{\neg,\not\subset\},\{T,\not\supset\},\{T,\not\subset\},\{F,\subset\},\{F,\supset\},\{\subset,\not\equiv\},\{\supset,\not\equiv\},\{\not\equiv,\equiv\},\{\subset,\not\supset\},\{\subset,\not\subset\},\{\supset,\not\supset\},\{\supset,\not\subset\},\{T,\vee,\not\equiv\},\{T,\&,\not\equiv\},\{F,\vee,\equiv\},\{F,\&,\equiv\},\{\vee,\equiv,\not\equiv\},$ and $\{\&,\equiv,\not\equiv\})$ and 4 maximally inadequate sets of connectives $(\{T,F,\vee,\&\},\{T,F,\neg,\equiv,\not\equiv\},\{T,\vee,\subset,\supset,\&,\equiv\},$ and $\{F,\vee,\&,\not\supset,\not\subset,\not\equiv\})$.

III. Frege Systems

A rule of inference is a pair (S,B), written $R = S \to B$, where S is a (possibly empty) finite set of formulas and B is a formula. R is said to be a rule in L_κ if B and all of the formulas in S are formulas in L_κ. Rule $R = S \to B$ is sound iff $S \models B$. Observe that for any substitution σ, $S \models B$ implies that $S_\sigma \models B_\sigma$. If $R = S \to B$ is sound and $S = \phi$, then all substitution instances of B are tautologies, and B (or R) is often called an axiom or axiom scheme. If $R = \{A_1,\ldots,A_k\} \to B$ is a rule of inference, and $C_1,\ldots,C_k,D$ are formulas, then D is inferred from $C_1,\ldots,C_k$ by R iff there is a substitution σ such that $C_1 = A_1\sigma,\ldots,C_n = A_n\sigma$, and $D = B\sigma$. If R is sound and D is inferred from $C_1,\ldots,C_k$ by R, then $C_1,\ldots,C_k \models D$.

A deduction system is a pair $I = (\kappa,R)$, where κ is a set of connectives and R is a finite set of sound rules of inference in L_κ. A derivation in deduction system $I = (\kappa,R)$ of formula B from the set of formulas S is a sequence $D = \langle A_1,\ldots,A_n\rangle$ of formulas in L_κ such that for each i, $1 \le i \le n$, A_i is inferred from formulas in $S \cup \{A_1,\ldots,A_{i-1}\}$ by some rule in R, and $A_n = B$. The notation $S \vdash_I A$ means there is a derivation in I of A from S, and $S \vdash_I A$ via D means D is a derivation in I of A from S.

Since $\models$ is a transitive relation, it follows that $S \models A$ whenever $S \vdash_I A$.

Deductive system I is complete iff for every valid formula A in L_κ, $\vdash_I A$.

System I is implicationally complete iff for every S,A in L_κ such that $S \models A$, $S \vdash_I A$. If D is a derivation, then $D\sigma$ is the sequence of formulas obtained by applying σ to each formula in D. Since rules of inference are transparent to substitution, if $S \vdash_I A$ via D and if σ is in L_κ, then $S\sigma \vdash_I A\sigma$ via $D\sigma$.

A Frege system is an implicationally complete deduction system $F = (\kappa,R)$ where κ is adequate. The notion of a Frege system is intended to describe the essential characteristics of the deductive systems found in most textbooks on mathematical logic. For example, one system described by Mendelson [1964] is a Frege system $M = (\kappa,R)$, where $\kappa = \{\neg,\supset\}$ and $R = \{\rightarrow(P\supset(P0\supset P)),\rightarrow((P\supset(P0\supset P1))\supset((P\supset P0)\supset(P\supset P1))),\rightarrow((\neg P\supset\neg P0)\supset(P0\supset P)),\{P,(P\supset P0)\}\rightarrow P0\}$. Frege's original system (Frege [1879]) had six axiom schemes and the rule modus ponens: $\{P,P\supset P0\} \rightarrow P0$. Other Frege systems can be found in Hilbert-Ackermann [1950], Kleene [1952,1967], Mendelson [1964], and Schoenfield [1967]. The interesting fact about Frege systems is that all Frege systems are super (when viewed as proof systems for tautologies) if and only if any one particular Frege system is super. This fact will be proved in the following development.

Let $F_1 = (\kappa_1,R_1)$ and $F_2 = (\kappa_2,R_2)$ be two Frege systems, and suppose $S \vdash_{F_1} A$ via D_1. In the cases where either κ_1 contains neither $\equiv$ nor $\not\equiv$ or κ_2 contains $\equiv$ or $\not\equiv$, there is a direct way of translating formulas in L_{κ_1} and L_{κ_2} such that td(A) (the direct translation of A) is logically equivalent to A, has length proportional to the length of A, and has approximately the same subformula structure as A. When such a translation exists, there is a derivation D_2, whose length is bounded by a constant times the square of the length of D_1, such that $td(S) \vdash_{F_2} td(A)$ via D_2. When κ_1 contains $\equiv$ or $\not\equiv$ and κ_2 does not contain either $\equiv$ or $\not\equiv$, there is no direct translation with the three necessary properties, so an indirect type of translation must be used. Whereas $td(A) \sim A$, ti(A) (the indirect translation of A) only has the property that ti(A) is valid if and only if A is valid. In this indirect case, all that can be said is that if $\vdash_{F_1} A$ via D_1, there is a derivation D_2, whose length is bounded by a constant times the fourth power of the length of D_1, such that $\vdash_{F_2} ti(A)$ via D_2.

To make the above informal discussion more precise, several notions of length will be defined. If A is a formula, then $|A|$ is the number of symbols in A, and $|A|_a$ is the number of occurrences of atoms in A. If $D = \langle A_1,\ldots,A_\ell\rangle$ is a derivation, then $|D| = \sum_{i=1}^{n} |A_i|$, $|D|_a = \sum_{i=1}^{n} |A_i|_a$, $|D|_\ell = \ell$, and $|D|_s = |\mathrm{sub}(D)|$, the number of distinct subformulas which occur in formulas in D.

Direct Translation

Let κ be any adequate set of connectives. Let $\kappa' = \kappa \cup \{T,F,\neg\}$, and let κ^+ = all connectives if κ contains $\equiv$ or $\not\equiv$, or all connectives except $\equiv$ and $\not\equiv$ if κ does not contain either $\equiv$ or $\not\equiv$. A direct translation from κ_1 to κ_2 is defined to be a function td : {formulas in L_{κ_1}} $\rightarrow$ {formulas in L_{κ_2}} which satisfies:

1. $td(A) \sim A$
2. $|td(A)| \leq c|A|$, for some constant c independent of A
3. $td(A\frac{B}{p}) = td(A)\frac{td(B)}{p}$
4. $\forall p \in at(A)$: the number of occurrences of p in td(A) equals the number of occurrences of p in A.

The following facts are easily verified for any adequate set of connectives κ.

a. There is a formula T_κ in L_κ with $T_\kappa \sim T$. (Let T_κ be the shortest tautology in L_κ. For example, $T_{\{|\}} = ((P|P)|P)$.)

b. There is a formula F_κ in L_κ with $F_\kappa \sim F$. (Let F_κ be the shortest inconsistent formula in L_κ. For example, $F_{\{|\}} = (((P|P)|P)|((P|P)|P))$.)

c. There is a formula N_κ in L_κ with only one occurrence of P0 such that $N_\kappa \sim \neg P0$. (For example, $N_{\{|\}} = (((P|P)|P)|P0)$.)

d. κ contains some v-like connective: $\circ$.

e. For every v-like connective x, there is a formula B_κ^x in L_κ, with one occurrence each of P0 and P1 such that $B_\kappa^x \sim (P0xP1)$. (If A^0 means A and A^1

means $\neg A$, then there are $0 \le \varepsilon_1,\varepsilon_2,\varepsilon_3 \le 1$ such that $(P0^{\varepsilon_1} \circ P1^{\varepsilon_2})^{\varepsilon_3} \sim (P0 x P1)$. For example, $(P0 \supset P1) \sim (P0 \mid \neg P1) = B^{\supset}_{\{|\}}$.)

f. If κ contains $\equiv$ or $\not\equiv$, then for any $\equiv$-like connective x, there is a formula B^x_κ in L_κ, with one occurrence each of $P0$ and $P1$ such that $B^x_\kappa \sim (P0 x P1)$. (This comes from the equivalences $(P0 \not\equiv P1) \sim \neg(P0 \equiv P1)$ and $(P0 \equiv P1) \sim \neg(P0 \not\equiv P1)$.)

Using these facts it is easy to show that the following recursive definition defines td'_κ to be a direct translation from κ^+ to κ':

$td'_\kappa(p) = p$ for any atom p

$td'_\kappa(T) = T$

$td'_\kappa(F) = F$

$td'_\kappa(\neg B) = \neg td'_\kappa(B)$

$td'_\kappa(A \circ B) = B^{\circ}_{\kappa} \dfrac{td'_\kappa(A)\ \ td'_\kappa(B)}{P0 \qquad P1}$.

Then td''_κ is a direct translation from κ' to κ, when td''_κ is defined by:

$td''_\kappa(p) = p$

$td''_\kappa(T) = T_\kappa$

$td''_\kappa(F) = F_\kappa$

$td''_\kappa(\neg B) = N_\kappa \dfrac{td''_\kappa(B)}{P0}$

$td''_\kappa(A \circ B) = td''_\kappa(A) \circ td''_\kappa(B)$.

Finally, since the composition of two direct translations is a direct translation, if td_κ is defined by $td_\kappa(A) = td''_\kappa(td'_\kappa(A))$, then td_κ is a direct translation from κ^+ to κ.

Theorem 2 Let $F_1 = (\kappa_1, R_1)$ and $F_2 = (\kappa_2, R_2)$ be Frege systems with $\kappa_1 \subseteq \kappa_2^+$. Then there is a constant c such that whenever $S \vdash_{F_1} A$ via D_1 there is a derivation D_2 such that $td_{\kappa_2}(S) \vdash_{F_2} td_{\kappa_2}(A)$ via D_2 where $|D_2|_n \le c|D_1|_n$ and $|D_2| \le c|D_1|_n \cdot (|D_1| + \sum_{A \in S} |A|) \le c(|D_1| + |S|)^2$.

Proof: The notion of translation is easily extended to the translation of inferences. Suppose $R = \{A_1,\ldots,A_n\} \to B$ is a rule of R_1. Then $A_1,\ldots,A_n \models B$, and since F_2 is implicationally complete, let $td_{\kappa_2}(A_1),\ldots,td_{\kappa_2}(A_n) \vdash_{F_2} td_{\kappa_2}(B)$ via D_R. If $\sigma = \dfrac{A_1,\ldots,A_k}{p_1,\ldots,p_k}$ is a substitution in L_{κ_1}, define $td_{\kappa_2}(\sigma) = \dfrac{td_{\kappa_2}(A_1),\ldots,td_{\kappa_2}(A_n)}{p_1,\ldots,p_n}$, so that $td_{\kappa_2}(A\sigma) = td_{\kappa_2}(A)td_{\kappa_2}(\sigma)$. Now to translate the given derivation D_1, for each formula A_i in D_1 which is inferred from previous formulas in D_1 and formulas in S by substitution of σ_i in rule R_i, replace A_i by $D_{R_i} td_{\kappa_2}(\sigma_i)$. The resulting derivation D_2 is a derivation such that $td_{\kappa_2}(S) \vdash_{F_2} td_{\kappa_2}(A)$ via D_2. Since $|D_R|_n$ is a fixed constant for each R, it is clear that $|D_2|_n \le c|D_1|_n$. If $\sigma = \dfrac{A_1,\ldots,A_k}{p_1,\ldots,p_k}$, define $|\sigma| = \sum_{i=1}^{k} |A_i|$. Since $|D_R|_a$ is a fixed constant for each R, there is a constant c_R such that $|D_R\sigma| \le c_R|\sigma|$. Now, if A_i is inferred in D_i by substitution of σ_i in rule R_i, then, since the number of atoms in R_i is fixed, and since each formula in σ_i must be a subformula of some formula in S or D_1, there is some constant c' such that $|\sigma_i| \le c'(|D_1| + \sum_{A \in S} |A|)$. Finally, since translation causes no more than linear expansion of formulas, summation of the relation $|D_{R_i}\sigma_i| \le c(|D_1| + \sum_{A \in S} |A|)$ for all formulas A_i in D_1 yields $|D_2| \le c|D_1|_n(|D_1| + \sum_{A \in S} |A|)$. Q.E.D.

Indirect Translation

If κ_1 contains $\equiv$ or $\not\equiv$ and κ_2 contains neither $\equiv$ nor $\not\equiv$, then the above type of direct translation will not work, since there can be no formula in L_{κ_2} with no more than one occurrence each of $P0$ and $P1$ which is logically equivalent to either $(P0 \equiv P1)$ or $(P0 \not\equiv P1)$. In this situation, the following indirect translation will suffice.

Let S be a set of formulas in L with $|S|_s = s$. To each distinct formula A in $sub(S)$ for which $\neg$ is not the principal connective, associate a unique atomic formula $\ell_S(A)$. This association can be made in such a way that $\ell_S(T) = T$, $\ell_S(F) = F$, and for each $A \in sub(S)$

$|\ell_S(A)| \leq 1 + \lfloor \log_2(s) \rfloor$. For formulas in $sub(S)$ of the form $\neg A$, set $\ell_S(\neg A) = \overline{\ell_S(A)}$, where $\overline{p} = N_\kappa \frac{p}{P0}$ and $\overline{N_\kappa \frac{p}{P0}} = p$. This assigns a literal (atom or negation of an atom) of length $|\ell_S(A)| \leq 1 + |N_\kappa| + \lfloor \log_2(s) \rfloor$ to each subformula in $sub(S)$, with distinct (after removal of double negations) formulas being assigned distinct literals, and complementary formulas being assigned complementary literals.

For each binary connective $\circ$, let E°_κ be a formula in L_κ such that $E^\circ_\kappa \sim (P \equiv (P0 \circ P1))$, and let $c'_\kappa = \max_\circ |E^\circ_\kappa|$. For each formula $A \circ B$ in $sub(S)$ whose principal connective is binary, define $\ell^\kappa_S(A \circ B) = E^\circ_\kappa \frac{\ell_S(A \circ B)\ \ell_S(A)\ \ell_S(B)}{P \quad P0 \quad P1}$, noting that $|\ell^\kappa_S(A \circ B)| \leq c'_\kappa(1 + |N_\kappa| + \lfloor \log_2(s) \rfloor) \leq c_\kappa \lfloor \log_2(s) \rfloor$. Finally, let $def^\kappa_S(S) = td_\kappa(\mathop{\&}_{(A \circ B) \in sub(S)} \ell^\kappa_S(A \circ B))$, where the parenthesization of the large conjunction is done in some standard way. (For example, the $(A \circ B)$'s are ordered lexicographically, and the grouping is from left to right: $((\ldots((E_1 \& E_2) \& E_3) \& \ldots) \& E_s)$.) If $C \in sub(S)$, then $def^\kappa_S(C) = td_\kappa(\mathop{\&}_{(A \circ B) \in sub(C)} \ell^\kappa_S(A \circ B))$. There is a constant d_κ such that $|def^\kappa_S(C)| \leq d_\kappa |C|_s \lfloor \log_2 |S|_s \rfloor$. Note that there is a renaming substitution p such that $def^\kappa_S(C) = def^\kappa_C(C)p$. Also, if $\sigma = \frac{A_1, \ldots, A_k}{p_1, \ldots, p_k}$, and $C = B\sigma$, then there is a renaming p such that $def^\kappa_S(C) \sim (def_B(B)p \& (\mathop{\&}_{i=1}^{k} def^\kappa_S(A_i)))$. That is, if C is a substitution instance of B, then the conjuncts of the definition of C are the conjuncts of the definitions of the formulas substituted into B plus a renaming of the conjuncts of the definition of C.

The following lemmas show how $def^\kappa_S(C)$ captures the important properties of C. Let p_S be the renaming induced by ℓ_S on the atoms of S, (i.e. $p_S = \frac{\ell_S(p_1) \ldots \ell_S(p_k)}{p_1 \ldots p_k}$, where $\{p_1, \ldots, p_k\}$ are the atoms of S.) Clearly Sp_S is is satisfiable (falsifiable) iff S is satisfiable (falsifiable). Also, ℓ_S can be chosen so that for every A in $sub(S)$ $|Ap_S| \leq |A|$, by choosing

$$\ell_S(p) = \begin{cases} p & \text{if } |p| \leq 1 + \lfloor \log_2(s) \rfloor \\ \text{some new } p' \text{ s.t. } |p'| \leq 1 + \lfloor \log_2(s) \rfloor & \text{otherwise} \end{cases}$$

Lemma 1 τ satisfies E°_κ iff $\tau(P) = \tau(P0 \circ P1)$.

Proof Immediate from the definition of E°_κ.

Lemma 2 τ satisfies $\ell^\kappa_S(A \circ B)$ iff $\tau(\ell_S(A \circ B)) = \tau(\ell_S(A) \circ \ell_S(B))$.

Proof Substitute definition of $\ell^\kappa_S(A \circ B)$ into lemma 1.

Lemma 3 τ satisfies $def^\kappa_S(S)$ iff $\forall A \in sub(S) : \tau(Ap_S) = \tau(\ell_S(A))$.

Proof

=>) Induction on the number of subformulas of A.

basis: A is an atomic formula. If A is atomic, then $Ap_S = \ell_S(A)$, so $\tau(Ap_S) = \tau(\ell_S(A))$.

induction step: Suppose $\tau(Ap_S) = \tau(\ell_S(A))$ and $\tau(Bp_S) = \tau(\ell_S(B))$. Then $(\neg A)p_S = \neg(Ap_S)$ and $(A \circ B)p_S = (Ap_S \circ Bp_S)$.

1) $\tau(\neg Ap_S) = \left\{ \begin{matrix} f & \text{if } \tau(Ap_S) = t \\ t & \text{if } \tau(Ap_S) = f \end{matrix} \right\}$

$= \left\{ \begin{matrix} f & \text{if } \tau(\ell_S(A)) = t \\ t & \text{if } \tau(\ell_S(A)) = f \end{matrix} \right\}$

$= \tau(\neg \ell_S(A)) = \tau(\ell_S(\neg A))$.

2) Since τ satisfies $def^\kappa_S(S)$, τ satisfies each of its conjuncts, $\ell^\kappa_S(A \circ B)$ in particular. By lemma 2, $\tau(\ell_S(A \circ B)) = \tau(\ell_S(A) \circ \ell_S(B))$, and by the induction hypothesis $\tau(\ell_S(A) \circ \ell_S(B)) = \tau((A \circ B)p_S)$, so $\tau((A \circ B)p_S) = \tau(\ell_S(A \circ B))$.

<=) Suppose that $\forall A \in sub(S)$: $\tau(Ap_S) = \tau(\ell_S(A))$. Then for each $(A \circ B) \in sub(S)$, $\tau((Ap_S \circ Bp_S)) = \tau(\ell_S(A \circ B))$, $\tau(Ap_S) = \tau(\ell_S(A))$, and $\tau(Bp_S) = \tau(\ell_S(B))$. By the definition of extension of a truth assignment, $\tau((Ap_S \circ Bp_S)) = \tau(\ell_S(A) \circ \ell_S(B))$, so that $\tau(\ell_S(A \circ B)) = \tau(\ell_S(A) \circ \ell_S(B))$. Then, by lemma 2, τ satisfies $\ell^\kappa_S(A \circ B)$. Finally, since τ satisfies each

conjunct of $def_S^K(S)$, τ satisfies $def_S^K(S)$.

Lemma 4 $A \in sub(S)$ is falsifiable iff $td_K(def_S^K(S) \supset \ell_S(A))$ is falsifiable.

Proof
=>) Suppose τ falsifies A.
Then $\tau' = \tau p_S$ falsifies Ap_S.
$((\tau\sigma)(B) = \tau(B\sigma))$
Extend τ' to τ'' which assigns arbitrary truth values to the atoms of Sp_S not in Ap_S. Then extend τ'' to τ''' such that $\forall B \in sub(S)$, $\tau'''(\ell_S(B)) = \tau''(Bp_S)$. Such an extension is possible, since each distinct $B \in sub(S)$ (after removal of double negations) has a distinct $\ell_S(B)$. Now, by lemma 3, τ''' satisfies $def_S^K(S)$ with $\tau'''(\ell_S(A)) = \tau'''(Ap_S) = f$. But this means that τ''' falsifies $td_K(def_S^K(S) \supset \ell_S(A))$.
<=) Suppose τ falsifies $td_K(def_S^K(S) \supset \ell_S(A))$.
Then $\tau(def_S^K(S)) = t$ and $\tau(\ell_S(A)) = f$. By lemma 3, $\tau(Ap_S) = \tau(\ell_S(A)) = f$. But then $\tau' = \tau p_S^{-1}$ (renamings are surely invertible) falsifies A.

Define an indirect translation to be $ti_K^S(A) = td_K(def_S^K(S) \supset \ell_S(A))$. Also define $ti_K^{S'}(S) = \bigcup_{A \in S} \{ti_K^{S'}(A)\}$

Theorem 3 Let $F_1 = (\kappa_1, R_1)$ and $F_2 = (\kappa_2, R_2)$ be two arbitrary Frege systems. Then there is a constant c such that whenever $S \vdash_{F_1} A$ via D_1 there is a set S' with $A \in S$ and a derivation D_2 such that $ti_{\kappa_2}^{S'}(S) \vdash_{F_2} ti_{\kappa_2}^{S}(A)$ via D_2 where $|D_2|_n \le c|D_1|_n \cdot |D_1 \cup S|_s$ and $|D_2| \le c|D_1|_n \cdot (|D_1 \cup S|_s)^2 \cdot \log_2(|D_1 \cup S|_s) \le c(|D_1 \cup S|)^4$.

Proof The proof of this theorem is based on the following two lemmas.

Lemma 5 Suppose $R = \{A_1, \ldots, A_k\} \to B$ is a sound rule of inference. Then $((p \& def_S^K(A_1)) \supset \ell_S(A_1)), \ldots, ((p \& def_S^K(A_k)) \supset \ell_S(A_k)), \vDash ((p \& def_S^K(B)) \supset \ell_S(B))$ where $S = \{A_1, \ldots, A_k\} \cup \{B\}$.

Proof Suppose τ satisfies $((p \& def_S^K(A_1)) \supset \ell_S(A_1)), \ldots, ((p \& def_S^K(A_k)) \supset \ell_S(A_k))$. 2 cases arise:

1) τ satisfies $def_S^K(S)$ and p.
Then, by lemma 3, $\tau(\ell_S'(A_1)) = \tau(A_1 p_S)$, $\ldots, \tau(\ell_S(A_n)) = \tau(A_n p_S)$, and $\tau(\ell_S(B)) = \tau(Bp_S)$. Since R is sound, $A_1, \ldots, A_n = B$, so $A_1 p_S, \ldots, A_n p_S \vDash Bp_S$. Since τ satisfies $def_S^K(S)$ and $ti_K^S(A_i)$ $1 \le i \le n$, τ satisfies $A_i p_S$ $1 \le i \le n$. But then τ satisfies Bp_S, since $A_1 p_S, \ldots, A_n p_S \vDash Bp_S$, so $\tau(\ell_S(B)) = t$ and τ satisfies $ti_K^S(B)$.

2) τ falsifies $((p \& def_S^K(S))$.
Then τ satisfies $(def_S^K(S) \supset \ell_S(B)) \sim ti_K^S(B)$.

$\therefore$ $ti_K^S(A_1), \ldots, ti_K^S(A_n) \vDash ti_K^S(B)$.

Lemma 6 For any Frege system $F = (\kappa, R)$ there exist constants c_1 and c_2 such that given any set S_1 of n formulas in L_κ, a set S_2 of m formulas from S_1, $C = td_K(\&_{A \in S_1} A)$, $D = td_K(\&_{A \in S_2} A)$, and atom p, there is a serivation D such that $td_K((C \& D) \supset p) \vdash_F td_K(C \supset p)$ via D and $|D|_n \le c_1 mn$ and for each formula $A \in D$ $|A| \le c_2|C|$.

Proof Since each conjunct of D is a conjunct of C, $C \vDash C \& D$.
The details of the rest of the proof are rather tedious but straightforward. In a Frege system F', designed for easy manipulations of conjunctions and implications, a derivation $D' : ((C \& D) \supset p) \vdash_{F'} (C \supset p)$ via D' with $|D'|_n \le cmn$ is derived. Then a direct translation of D' yields the desired derivation D.

Now, to prove the indirect translation theorem, set $S' = S \cup D$ and set $D = def_{\kappa_2}^{S'}(S')$. Corresponding to each formula B in D_1, D_2 contains a derivation of $ti_{\kappa_2}^{S'}(B)$. To be specific suppose that in D_1, B is inferred from $A_1, \ldots, A_k$ by substitution σ in rule $R = \{A_1', \ldots, A_k'\} \to B'$. Thus $A_1 = A_1'\sigma, \ldots, A_n = A_k\sigma$, and

$B = B'\sigma$. As induction hypotheses, assume $S \cup D_2$ already contains $ti^{S'}_{\kappa_2}(A_1),\ldots,$ $ti^{S'}_{\kappa_2}(A_k)$. Let D_R be the derivation whose existence is implied by lemma 5. Let $S^R = \{A'_1,\ldots,A'_k,B'\}$. Then, as was previously observed, there is a renaming p such that every conjunct of $def^{\kappa}_{S^R}(A'_i)p$ is a conjunct of $def^{\kappa}_{S'}(S')$. Thus

$$td_\kappa(((def^{\kappa}_{S'}(S')\&def^{\kappa}_{S^R}(A'_1)p)\supset \ell_{S'}(A_1))),$$
$$\ldots,td_\kappa(((def^{\kappa}_{S'}(S')\&def^{\kappa}_{S^R}(A_k)p)\supset \ell_{S'}(A'_k)))$$
$$\vdash_{F_2} td_\kappa(((def^{\kappa}_{S'}(S')\&def^{\kappa}_{S^R}(B)p)\supset \ell_{S'}(B)))$$

via $D_R p\ \dfrac{def^{\kappa}_{S'}(S')}{p}$.

Clearly, since $(A\supset P) \models ((A\&B)\supset P)$, there are derivations in F_2 of $td_\kappa(((def^{\kappa}_{S'}(S')\&def^{\kappa}_{S^R}(A'_i)p)\supset \ell_{S'}(A_i)))$ from $td_\kappa((def^{\kappa}_{S'}(S')\supset \ell_{S'}(A_i))$ each with a fixed number of lines proportional in length to $|def^{\kappa}_{S'}(S')|$. By lemma 6, there is a derivation of $td_\kappa((def^{\kappa}_{S'}(S')\supset \ell_{S'}(B))$ from $td_\kappa(((def^{\kappa}_{S'}(S')\&def^{\kappa}_{S^R}(B)p)\supset \ell_{S'}(B)))$ which has no more than $c_1|S'|_s|S^R|_s$ lines of length bounded by $c_2|S'|_s\lfloor\log_2|S'|_s\rfloor$. Since D_R is fixed and $|S^R|_s$ is fixed, this entire derivation that serves in D_2 to replace B has $\leq c|S'|_s$ lines each of length $\leq c|S'|_s\log|S'|_s$. Finally, summing these bounds for each formula B in D_1,

$$|D_2|_n \leq c|D_1|_n|D_1\cup S|_s \quad \text{and}$$
$$|D_2| \leq c|D_1|_n(|D_1\cup S|_s)^2\log_2(|D_1\cup S|_s)$$
$$\leq c(|D_1|+|S|)^4.$$

IV. Other Powerful Proof Systems

Other systems that have been proposed for proving validity or unsatisfiability of propositional formulas include natural deduction (Fitch [1952], Kleene[1967]), Gentzen systems (Wang [1960], Kleene [1967]), and extended resolution (Tseitin [1968]). All of these types of systems turn out to be equivalent to Frege systems in the sense that any one such system is super if and only if all such systems and all Frege systems are super.

Natural Deduction

Natural deduction systems are often derived from Frege systems by the addition of a derived rule, the deduction theorem: if $A_1,\ldots,A_n \vdash B$ then $A_1,\ldots,A_{n-1} \vdash A_n\supset B$.

All Frege systems have a deduction theorem, so each Frege system gives rise to a natural deduction system. A line in a natrual deduction system is a string of the form $A_1,\ldots,A_n \vdash B$, where the A_i and B are formulas. Such a line is considered equivalent to $((A_1\&\ldots\&A_n)\supset B)$. A rule of inference in such a system has the form $\{\ell_1,\ldots,\ell_k\} \to \ell'$, where the ℓ_i and ℓ' are lines of the form $\Delta,A_1,\ldots,A_k \vdash B$. ($\Delta$ appears in every line of the rule.) The rule is sound iff $\ell_1,\ldots,\ell_k \models \ell'$. To obtain an instance of such a rule, apply any substitution σ to $A_1,\ldots,A_k,B$, and replace Δ by any list of formulas. A derivation in such a system is a sequence of lines, each of which is obtained from previous lines (or hypotheses) by a rule of inference.

Every Frege system is a natural deduction system, if we just read rule $\{A_1,\ldots,A_n\} \to B$ as $\{\Delta\vdash A_1,\ldots,\Delta\vdash A_n\} \to \Delta\vdash B$. In fact, in such a system, nothing will ever be substituted for Δ.

Theorem 4 If N and N' are implicationally complete natural deduction systems, there is a polynomial p such that for any derivation D in N there is a derivation D' in N' of a translation of the conclusion of D from the translation of the hypotheses of D of length $|D'| \leq p(|D|)$.

Proof The proof of this theorem follows along the same lines as previous proofs. Each line of D is simulated by a short derivation of the translation of that line.

Gentzen Systems

We shall outline a version of these systems as adapted from Kleene [1967]. A sequent is an ordered pair (Δ,Λ) of sets of formulas, written $\Delta \to \Lambda$. We shall assume that Δ and Λ are true sets, so that order of occurrence and multiplicity of occurrence in Δ and Λ have no meaning. A sequent $A_1,\ldots,A_m \to B_1,\ldots,B_n$ is valid iff the formula $(A_1\&\ldots\&A_m)\supset(B_1\vee\ldots\vee B_n)$ is valid, and the sequent takes on the same truth value under a truth assignment ϕ as the formula $(A_1\&\ldots\&A_m)\supset(B_1\vee\ldots\vee B_n)$. Corresponding to each of the connectives $\neg,\&,\vee,\supset,\equiv$ there are two rules of inference, one for introduction and one for elimination of the connective. The rules for $\neg$ and $\supset$ are

	introduction	elimination
$\neg$	$\dfrac{A,\Gamma \to \Theta}{\Gamma \to \Theta,\neg A}$	$\dfrac{\Gamma \to \Theta,A}{\neg A,\Gamma \to \Theta}$
$\supset$	$\dfrac{A,\Gamma \to \Theta,B}{\Gamma \to \Theta,A\supset B}$	$\dfrac{\Gamma \to \Theta,A \quad B,\Gamma \to \Theta}{A\supset B,\Gamma \to \Theta}$

Here upper case Greek letters stand for sets of formulas, upper case latin letters stand for formulas, and the notation A,Γ, for example, means $\{A\} \cup \Gamma$.

The final rule is called

cut: $\dfrac{\Delta \to \Lambda,C \quad C,\Gamma \to \Theta}{\Delta,\Gamma \to \Lambda,\Theta}$

This rule is not necessary for completeness, but it seems to allow proofs to be considerably shortened in general.

The Gentzen axioms are sequents of the form $C,\Gamma \to \Theta,C$.

A proof of a sequent $\Delta \to \Lambda$ in a Gentzen system is usually defined to be a rooted tree whose nodes are labelled with sequents, whose root is labelled with $\Delta \to \Lambda$, whose leaves are labelled with axioms, and such that each internal node is a consequence by a rule of inference of its daughter node(s). For our purposes, it is better to assume that either the proof has a linear format or that it is an acyclic digraph, so that once an intermediate sequent in a proof has been derived, it does not have to be derived again if it is used twice.

A proof of a formula A in a Gentzen system is a proof of the sequent $\to A$, and a proof that A is inconsistent is a proof of the sequent $A \to$.

The idea of a Gentzen system can be extended to any system with rules for deriving sequents. In this sense, natural deduction systems are Gentzen systems in which the second set Λ of the sequent $\Delta \to \Lambda$ is always a singleton set, and Frege systems are Gentzen systems where Δ is empty and Λ is a singleton.

A Gentzen system that is implicationally complete is called a Gentzen system with cut. Note that the above system is complete (for $\kappa = (\neg,\supset)$) without the cut rule, but that it is not implicationally complete unless the cut rule is included.

Theorem 5 Let G be a Gentzen system, and let G' be a Gentzen system with cut. Then there is a polynomial $p(\cdot)$ such that given any derivation D in G, there is a derivation D' in G' of the translation of the conclusion of D from the translations of the hypotheses of D such that $|D'| \leq p(|D|)$.

Proof The proof of this theorem follows the previous pattern. Each line of D is replaced in D' by a short derivation in G' of the translation of that line.

Resolution with Extension

An operation called consensus was introduced by Quine [1955] as a method to help find the minimum normal disjunctive form for a formula. It was adapted by Dunham [1962] as a computer method for establishing the validity of formulas in normal disjunctive form.

Resolution was introduced by Robinson [1965] as a proof method for the predicate calculus. When resolution is restricted to the propositional calculus, it is just the dual of consensus, and provides a method for establishing the inconsistency of a formula in normal conjunctive form. In resolution terminology, a literal is an atom or a negation of an atom, and a clause is a finite disjunction of literals. We will think of a clause as a set of literals, so that ordering and repetitions of literals in a clause does not make sense. The complement of a literal ξ is denoted by $\overline{\xi}$, and is defined by $\overline{p} = \neg p$, $\overline{\neg p} = p$. The resolvent of clauses $A \vee \xi$ and $\overline{\xi} \vee B$ is $A \vee B$, where it is understood that if a literal occurs in both A and B, the two occurrences are merged in $A \vee B$. A resolution refutation of a set S of clauses is a finite sequence of clauses each of which is either a number of S or a resolvent of two earlier clauses in the sequence, and such that the last clause is the empty clause $\Box$. One can prove that a set S of clauses is inconsistent if and only if S has a resolution refutation.

Tseitin [1968] introduced a rule, called extension, to be used in conjunction with extension, and which seems to allow for considerably shorter refutations of some sets of clauses. This rule allows the introduction of new atoms, and new clauses which force the new atoms to be equivalent to any truth-function of the original atoms.

Extension Rule: (for a set S of clauses). If α is a literal such that neither α nor $\overline{\alpha}$ occurs in S, then the three clauses $\alpha \vee \overline{\beta} \vee \overline{\gamma}$, $\overline{\alpha} \vee \beta$, $\overline{\alpha} \vee \gamma$ may be added to S, for any literals β and γ.

Notice that the conjunction of these clauses is equivalent to the formula $(\alpha \equiv (\beta \& \gamma))$.

The extension rule may be applied any number of times to a set S of clauses, provided the new variable α introduced is distinct for each application. The resulting augmented set is consistent if and only if the original set S is consistent. Thus a refutation using resolution with extension consists of a sequence of

extensions followed by a sequence of resolvents.

Resolution with extension can be made into a proof system for tautologies A with any connectives by using the methods described under the section on indirect translation of Frege systems. If we let $\kappa = \{\neg, \&, \vee\}$ and $S = \{A\}$, then $e_S^\kappa(B \circ C)$ may as well be in conjunctive normal form, and hence can be regarded as a set of clauses; (there will be at most four clauses, with at most three literals per clause). Then let def(A) be the union of $e_S^\kappa(B \circ C)$ over all subformulas $(B \circ C)$ of A. Then def(A) is a set of clauses with the property that $\text{def}(A) \models \ell(A)$ iff A is a tautology. Further, there exists a derivation using resolution of the unit clause $\ell(A)$ from the clauses def(A) iff A is a tautology. If we allow use of the extension rule in the derivation, then we will regard such a derivation D as a proof of A using resolution with extension. The length $|D|$ of D is the total number of symbols encoding the clause of the proof, under a suitable encoding.

Theorem 6 There is Frege system F and a constant c such that any proof D of a formula A in resolution with extension can be transformed to a proof D' of A in F with $|D'| \leq c|D|^3$, and the same with "F" and "resolution with extension" interchanged.

The idea of the proof is this. The system F will include all connectives, and rules will be suitable for simulating resolution with extension. A proof D of A using resolution with extension is a sequence of clauses representing a derivation of $\ell(A)$ from def(A). If we now substitute for each literal in each clause the formula defined by that literal (i.e. substitute B for $\ell(B)$), the result will be a sequence of tautologies ending in A. This can be expanded into a proof in F by including axiom schemas in F which make the substitution instances of $e_S^\kappa(B \circ C)$ just defined into instances of axioms. The other rules needed in F are the rule "cut", $\{A \vee \neg B,\ B \vee C\} \to A \vee C$, to simulate resolution, and rules implementing the associativity and commutativity of $\vee$, and double negation cancellation.

Conversely, given a proof $D = \langle A_1, \ldots, A_n \rangle$ of A in F, a derivation of $\ell(A)$ from def(A) can be constructed by using the extension rule to introduce literals $\ell(B)$ for every subformula B of every A_i such that B is not a subformula of A. Clauses defining literals associated with the subformulas of A are already available in def(A). Then the sequence of unit clauses $\ell(A_1), \ldots, \ell(A_n) = \ell(A)$ can be expanded into a derivation using resolution with extension.

We do not know whether a result similar to the theorem holds when "resolution with extension" is replaced by "resolution", but the result of Tseitin on regular resolution (described later) casts doubt on such a proposition.

V. Weaker Systems and Lower Bounds

Analytic Tableaux. These were introduced by Smullyan, and described in Smullyan [1968]. They were inspired by Gentzen systems and other similar systems, and provide a proof system which is elegant in principle, and satisfying to apply on short examples. However, since a tableau proof is defined to be a labelled tree, and since it cannot so readily be put in a linear form or the form of an acyclic digraph, examples can be found which require much duplication in the construction of a tableau proof. Here is a class of hard examples: Let $T_m = \{\pm P^1 \vee \pm P^2_{\pm} \vee \pm P^3_{\pm\pm} \vee \ldots \vee P^m_{\pm\ldots\pm}\}$ where $+P$ means P and $-P$ means $\neg P$, and the subscript of P^i is a string of i-1 + or -'s corresponding to the sequence of signs of the preceding p^j, $j < i$. Thus T_m is a set of 2^m clauses and has $2^0 + 2^1 + \ldots + 2^{m-1} = 2^m - 1$ atoms. For example, $T_2 = \{P^1 \vee P^2_+,\ P^1 \vee \neg P^2_+,\ \neg P^1 \vee P^2_-,\ \neg P^1 \vee \neg P^2_-\}$. If we let S_m be the conjunction of all disjunctions in T_m, then S_m has length about $(2m)2^m$, but one can show that any closed tableau for S_m has at least $2^{2^{cm}}$ nodes, where $c > 0$ is some constant independent of m. (We remark that there is a resolution refutation of T_m with length only 2^m-1.)

Davis-Putnam Procedure. Davis and Putnam [1960] introduced a decision procedure which amounts to determining whether a set of clauses is consistent. This procedure can be formulated in terms of resolution as follows:

> Select a literal ξ from the set S of clauses. From all possible resolvents of pairs of clauses from S in which ξ is the literal resolved upon (i.e. all pairs of the form $A \vee \xi$, $\overline{\xi} \vee B$). If the empty clause is thus formed, S is inconsistent. From the set of resolvents, delete all clauses which have both a literal and its complement, and let S' be the remaining resolvents, together

with all members of S which contain neither ξ nor $\overline{\xi}$. If S' is empty, then S is consistent. Now repeat the whole procedure (for a different literal ξ) for S' instead of S. (Note that S' is consistent if and only if S is consistent, and S' contains one less atom than S.)

Thus we can think of the Davis-Putnam procedure as a procedure for generating resolution proofs of a certain restricted form. In Cook [1971b], examples are given which show the length of the resulting proofs depend heavily on the order in which the literals ξ are eliminated.

Also, the Davis-Putnam procedure as originally stated, and as stated above, omits "subsumption". This rule allows us to delete every clause A which is a super set of another clause B also present. Subsumption can be applied before applying the Davis-Putnam procedure, and it can be applied during the procedure every time a resolvent is generated. Examples given in Cook [1971b] and Simon [1971] for which there is no polynomial bound on the number of clauses generated by the Davis-Putnam procedure, no matter in what order the atoms are eliminated. However, if the subsumption rule is applied, all the examples generated only a few clauses under Davis-Putnam. The status of the Davis-Putnam procedure is definitely settled by a result in Tseitin [1968].

Regular Resolution. Tseitin [1968] defines a resolution refutation to be irregular iff there is some subsequence of formulas $A_1,\ldots,A_k$ from the proof such that each A_i is one of the parent clauses in forming the resolvent A_{i+1}, $1 \le i \le k-1$, and such that there is some literal ξ which appears in A_1 and A_k, but is not present in some intermediate clause A_i. Thus ξ is removed by resolution, and then reintroduced by another branch of the proof (so it must be moved again if the empty clause is formed). The refutation is regular if it is not irregular.

Theorem 7 (Tseitin) For infinitely many integers n there is an inconsistent set of n clauses whose minimum regular resolution refutation has length exceeding $2^{\sqrt{n}}$.

The proof, which appears in Tseitin [1968], is long but elegant.

Corollary: For infinitely many integers n there is an inconsistent set of n clauses such that the Davis-Putnam procedure with subsumption generates at least $2^{\sqrt{n}}$ clauses, no matter in what order the atoms are eliminated.

The corollary shows that neither the Davis-Putnam procedure, nor any obvious variation of the procedure, is a super proof system. It also shows that no obvious variation is a polynomial time decision procedure. The proof of the corollary amounts to the easy observation that every Davis-Putnam refutation (even with subsumption) correspond to a regular resolution refutation.

It is not known whether theorem 7 applies to resolution in general (as opposed to regular resolution).

Semantic Trees. These were discussed in Robinson [1968] and Kowalski-Hayes [1969] as a general method for analysing mechanical proof procedures for the predicate calculus. We shall present a special version useful as a refutation system for the propositional calculus.

A partial truth assignment to a set S of formulas is a map $Q : at(S) \to \{t,f,u\}$ (u is for "undefined"). The map Q can be extended to $sub(S)$ by the recursive equations $Q(T) = t$, $Q(F) = f$,

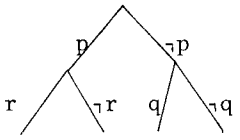
$$Q(\neg A) = \begin{cases} t & \text{if } Q(A) = f \\ f & \text{if } Q(A) = t, \\ u & \text{if } Q(A) = u \end{cases}$$

$Q((A \circ B)) = Q(A) \circ Q(B)$, where $Q(A) \circ Q(B)$ is given by the truth table in section II if $Q(A) \neq u$ and $Q(B) \neq u$, and if just one of $Q(A)$, $Q(B) \neq u$, then $Q(A) \circ Q(B)$ is t or f if this can be determined without knowing the value of the other, and otherwise $Q(A) \circ Q(B) = u$. For example, if $\circ$ is $\&$, then $f\&u = f$, $u\&f = f$, but $t\&u = u\&t = u\&u = u$.

A semantic tree for a set S of formulas is a finite binary rooted tree, with the pair of edges leading out from each node labelled P and $\neg P$ respectively, for some P in $at(S)$, and such that no branch (i.e. path from root to a leaf) has a complementary pair of literals on it. A branch determines a partial truth assignment Q by the conditions $Q(p) = t$ if p labels some edge on the branch, $Q(p) = f$ if $\neg p$ is on the branch, and $Q(p) = u$ if neither p nor $\neg p$ is on the branch. The branch is closed for S iff the partial truth assignment determined by the branch falsifies some formula is S. The tree is closed for S iff every branch is closed.

Theorem 8 A set S of formulas is inconsistent iff there is some closed semantic tree for S.

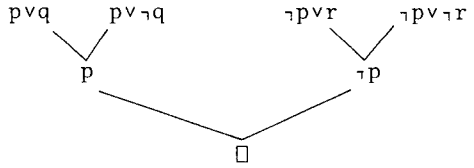
The proof can be adapted from results in Robinson [1968]. An example of a closed semantic tree for $S = \{p \vee q, p \vee \neg q, \neg p \vee r, \neg p \vee \neg r\}$ is



Tree Resolution and Semantic Trees

A tree resolution refutation of a set S of clauses is a finite binary rooted tree whose nodes are labelled with clauses, such that the leaves are labelled with clauses from S, the root is labelled with the empty clause $\square$, and the clause labelling any internal node is the resolvent of the clauses labelling the two parent nodes of the node. It follows from the completeness of resolution that S has a tree resolution refutation iff S is inconsistent. However, the number of nodes on the smallest tree resolution refutation of S might be much greater than the number of nodes in an (ordinary) resolution refutation of S, because no resolvent formed on the tree can be used more than once.

Here is a tree resolution refutation for the set $S = \{p \vee q, p \vee \neg q, \neg p \vee r, \neg p \vee \neg r\}$ mentioned above



Theorem 9 Every closed semantic tree for a set of clauses S can be converted to a tree resolution refutation (of the same size and shape) by suitably changing labels, and conversely.

The ideas needed for the proof are in Robinson [1968] and Kawalski-Hayes [1969]. As a result of this, we can conclude the following.

Theorem 10 There is a constant $c > 0$ such that for infinitely many n there is an inconsistent set of n clauses whose smallest tree resolution refutation and smallest closed semantic tree both have at least $2^{c(\log_2 n)^2}$ nodes.

The result for tree resolution refutations follows from theorem 4 of Tseitin [1968], and the result for closed semantic trees follows from the theorem above.

Other Simulation Results

Theorem 11 For every closed analytic tableau T for a set S of clauses there is a closed semantic tree S with at most twice as many nodes as T.

Proof Using a depth-first search of the closed tableau, construct a semantic tree so that corresponding to every open path of the tableau which has literals $L_1,\ldots,L_k$ labelling nodes, the semantic tree will have (among other labels) edges labelled $L_1,\ldots,L_k$. Every edge L will have a mate labelled $\overline{L}$, which accounts for the factor of two. The semantic tree branches will always close before the tableau branches.

A very strong counter-example to the converse of this theorem is provided by the sets T_m described in section on analytic tableaux. These sets are exponential for tableaux, but linear for semantic trees.

Theorem 12 (Tseitin) There is a constant c such that given a cut-free proof of $\rightarrow A$ Gentzen's system $\mathfrak{L}$ of n lines there is a resolution refutation (without extension) of $def(A) \cup \{\overline{\mathfrak{L}(A)}\}$ of length at most cn.

The proof is outlined in section 1 of Tseitin [1968].

REFERENCES

Bauer et. al. [1973]. A note on disjunctive form tautologies, by Bauer, Brand, Fischer, Meyer, and Paterson. SIGACT NEWS, April, 1973.

Cook [1971a]. The complexity of theorem-proving procedures, by S.A. Cook. Proceedings of Third Annual ACM Symposium on Theory of Computing, May, 1971.

Cook [1971b]. Examples for the Davis-Putnam Procedure, by S.A. Cook. Unpublished manuscript, June, 1971.

Davis-Putnam [1960]. A computing procedure for quantification theory, by Martin Davis and Hilary Putnam. JACM, vol. 7, pp. 201-215.

Dunham [1962]. Theorem testing by computer, by B. Dunham and J.H. North. Proceedings of the Symposium on Mathematical Theory of Automata, Jerome Fox, Editor, pp. 173-177.

Frege [1879]. Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens, by G. Frege, Halle, 1879.

Hilbert-Ackermann [1950]. Principles of Mathematical Logic, by D. Hilbert and W. Ackermann, New York (Chelsea Pub. Co.).

Karp [1972]. Reducibility among combinatorial problems, by R.M. Karp, Complexity of Computer Computations, R.E. Miller and J.W. Thatcher, ed., New York (Plenum Press), pp. 85-103.

Kowalski-Hayes [1969]. Semantic trees in automatic theorem proving, by R. Kowalski and P. Hayes. Machine Intelligence, Vol. 4 (B. Meltzer and D. Michie, eds.), New York, pp. 87-101.

Kleene [1952]. Introduction to Metamathematics, by S.C. Kleene, (D. Van Nostrand, Inc.).

Kleene [1967]. Mathematical Logic, by S.C. Kleene, (Wylie).

Mendelson [1964]. Introduction to Mathematical Logic, by Elliott Mendelson, (Van Nostrand).

Quine [1955]. A way to simplify truth functions, by W.V. Quine, American Mathematical Monthly, Vol. 62, pp. 627-631.

Robinson [1965]. A machine oriented logic based on the resolution principle, by J.A. Robinson, JACM, Vol. 12, pp. 23-41.

Robinson [1968]. The generalized resolution principle, by J.A. Robinson. Machine Intelligence, Vol. 3 (D. Michie, ed.), American Elsevier, New York, pp. 77-94.

Simon [1971]. On the time required by the Davis-Putnam tautology recognition algorithm, by Imre Simon, Research Report CSRR-2050, Department of Applied Analysis and Computer Science, University of Waterloo, Waterloo, Ontario, Canada, June, 1971.

Smullyan [1968]. First order logic, by Raymond M. Smullyan, Springer-Verlag New York Inc.

Tseitin [1968]. On the complexity of derivation in propositional calculus, by G.S. Tseitin, Studies in Constructive Mathematics and Mathematical Logic, Part II, A.O. Slisenko, ed.

Wang [1960]. Toward Mechanical Mathematics, by Hao Wong. IBM Journal, Jan. 1960, pp. 2-22.