CS 99

PB 179057

LECTURE NOTES ON

FOUNDATIONS FOR COMPUTER SCIENCE

BY

JOYCE FRIEDMAN

TECHNICAL REPORT NO. CS 99
JUNE 11, 1968

COMPUTER SCIENCE DEPARTMENT
School of Humanities and Sciences
STANFORD UNIVERSITY

PREFACE

This report consists of notes prepared for Applied Mathematics ~~206b~~ at Harvard University in the Spring term of 1965 and Computer Science 208 at Stanford University in the Fall quarter of 1966 and 1967. They do not pretend ~~to be~~ more than lecture notes; ~~in particular~~, no attempt has been made to expand outlines and remarks into full sentences. In spite of ~~the~~ deficiences and ~~the~~ incompleteness ~~of the notes~~, students seem to find them useful. For this reason, they are reprinted as a technical report.

Mendelson's Introduction to Mathematical Logic, van Nostrand, 1964, was used as a supplementary text for the course. The formal treatment of the propositional calculus here is primarily a commentary on the test and is therefore incomplete.

Two sections of the notes are reprints of material written by others. The section on the Infinity Lemma is a translation by Anthony Sholl of a chapter of Konig's Theorie der Graphen, Chelsea, 1950, which is otherwise unavailable in English. Also included is the chapter "A very elementary system L" reprinted with minor changes from Hao Wang's A Survey of Mathematical Logic, Science Press, Peking, 1963.

CONTENTS

	Page
Historical Background	1
Introduction to the Propositional Calculus	4
Comment on the Form of the Truth-table	8
Tautologies	18
Normal Forms in the Propositional Calculus	25
Decision Methods in the Propositional Calculus (Quine, Davis-Putnam; Dunham)	30
An Axiom System for the Propositional Calculus	37
Historical Notes	42
First Order Predicate Calculus (Formal Development, Proof Procedures, wffs of Predicate Calculus)	45
Formal Theorems (Consistency, Deduction, Useful Theorem Schemata)	52
Equivalence Theorems	62
Prenex Normal Form	65
A Very Elementary System W (Hao Wang)	69
Satisfiability	76
Proof Procedures and Decision Procedures	81
Details of Soundness Proof	83
Skolem Normal Form	89
The Infinity Lemma (König's Infinity Lemma)	98
Gödel's Completeness Theorem	105
Set Theory	118
Naive Set Theory	119
Axiomatic Set Theory	127

CONTENTS (continued)

	Page
Turing Machines	134
Algorithms	135
Invariance of Turing Machines	145
The Halting Problem	152
Partial Recursive Functions	159
Shepherdson - Sturgis Machines	163
Normal Systems	176
Post Correspondence Problem	179
Ambiguity Problem for Context-free Grammars	185
The Domino Problem	186
Constrained Domino Problems	194
The Decision Problem for Predicate Calculus	209

BLANK PAGE

COMPUTER SCIENCE 208

FOUNDATIONS FOR COMPUTER SCIENCE

The short title for this course is "Foundations for Computer Science". A longer and more accurate title would be "Foundations of Mathematics for Computer Scientists: an introduction of logic, set theory, algorithms and computability."

Historical background

The historical background for a combined study of logic, algorithms and computation can be said to date back to 1666 and to Leibnitz.. who dreamed that some day philosophical and mathematical arguments could be avoided by calculation. He envisaged a universal characteristic, "a general method in which all truths of the reason would be reduced to a kind of calculation. At the same time this would be a sort of universal language or script, but infinitely different from all those projected hitherto; for the symbols and even the words in it would direct the reason; and errors, except those of fact, would be mere mistakes in calculation." Instead of disputing, men would simply calculate.

This dream, in a much more sophisticated form, was shared by Hilbert at the turn of this century. Hilbert emphasized that mathematics should be treated as a formal system, abstracted from its meaning. The study of the formal system was called metamathematics, or proof theory. One of the main problems of the Hilbert program was the decision problem--the problem of finding a general method to determine if a given mathematical statement is true. The mathematical problem would be expressed in terms of the formal system, and a purely mechanical procedure would determine if the conclusion was in fact a theorem of the formal system. At the time it did not occur to anyone that this would be impossible, although clearly the problem was difficult.

These hopes were destroyed by the work of Gödel, Turing and Church, who showed that it was impossible to find such general methods, even in quite restricted areas. (There are some very simply stated unsolvable problems.) Thus, the situation is that in the 1930's, 10 to 20 years before the hardware is ready, before the introduction and widespread use of modern high-speed computing machines, the dream of using mechanical methods to solve all of mathematics is shattered.

The reaction to these discoveries was violent. Von Neumann is said to have received word of Gödel's results while lecturing on logic He read the message, remarked "Gentlemen, I have nothing further to say," left, and never returned either to that class or to his work on logic and set theory

The discouragement because of the negative results on solvability has also been reflected in the actual use of computers. In the development and application of computers, the emphasis has been on numerical methods. This, of course, is primarily because of the rapid development of computers during the war, in response to a need for numerical computation, for ballistics, for atomic energy But it also reflects the tremendous discouragement of logicians by the great unsolvability results. As we shall see, it is not until the late 1950's that the question of using computers to apply decision procedures where they do exist, and of working out partial procedures in other cases is seriously considered.

Our treatment of logic, algorithms, and computation will be from both the negative and positive points of view. On the one hand, we will try to give a clear description of the limits of what cannot be computed, or solved by computation. On the other hand, we will investigate the areas in which computation can, either partially or completely, succeed.

BLANK PAGE

Introduction to the Propositional Calculus

We begin informally, introducing the propositional calculus with its usual application to sentences, by means of truth-table definitions for its connectives. (Later, we shall treat the propositional calculus as a formal system, and show by means of soundness and completeness proofs that the informal system is correctly and adequately described by the formal one.)

In the informal treatment the basic units are sentences, statements, or propositions. These are atomic indivisible units, and are declarative statements which admit to being either true or false. 7 is a prime. Harvard University is in Cambridge, Massachusetts. 4 is an odd number. These basic units are combined by means of connectives to form new compound sentences. If 4 is an odd number, then 7 is a prime. Harvard is in Cambridge if and only if 7 is a prime. The truth or falsehood (briefly, the truth-value) of the resulting combination then depends only on the truth-values of the component sentences, and not on any internal relation between them. Same two compound sentences. Consider also: Socrates is a man. All men are mortal. No conclusion in the propositional calculus. Socrates is a man. If Socrates is a man, then Socrates is mortal. Conclusion in the propositional calculus.

If we interpret the elements of the propositional calculus as sentences, we will wish also to interpret the

connectives as English words or phrases. There is a correspondence between the connectives and the words <u>not</u>, <u>and</u>, <u>or</u>, <u>if ... then</u>, and <u>if and only if</u>, which are usually used as translations. But the meaning of the connectives is a defined meaning, not subject to the various alternative readings which are available for the corresponding words. In some cases, the meaning of the connectives may appear unnatural, relative to the corresponding English, but since the connectives are precisely defined, no real damage can result.

The elementary or atomic statements of the propositional calculus are denoted by the <u>statement letters</u> p, q, r, p_1, q_1, r_1, ... The two truth-values are <u>falsehood</u>, denoted by '0' or by 'F', and <u>truth</u>, denoted by '1' or by 'T'. In general, no confusion arises from using numerals to denote the truth-values, and it is more convenient for computation. However, in proofs, the letters 'T' and 'F' will be used.

<u>The connectives</u>. Atomic sentences are combined into compound sentences by means of the connectives.

<u>Negation</u>.

There is one singulary connective, <u>not</u>, which corresponds to negation. The negation of p is usually written '$\bar{p}$' or '$-p$' or '$\neg p$' or '$\sim p$'. Its value, which depends only on the value of p, is given by the truth-table

p	0	1
-p	1	0

That is to say, -p is true if p is false, and -p is false if p is true.

While '-p' is generally read 'not p', it is, of course, not true that we form the negation of a sentence in English simply by prefixing the word 'not'. '7 is not a prime' is the negation of '7 is a prime' The word 'not' is thus placed within the sentence. To obtain a uniform method of translation of -p in terms of p we may use 'it is not the case that p ', <u>it is not the case that 7 is a prime</u>.

There are four binary connectives, also defined by truth-tables.

<u>Conjunction</u>.

The truth-table for conjunction, p <u>and</u> q is

p	0	1	0	1
q	0	0	1	1
(p ∧ q)	0	0	0	1

(p ∧ q) is true only when <u>both</u> p is true <u>and</u> q is true. (p ∧ q) is also written 'p & q', and 'p · q' and 'pq'.

$p \wedge q$ is the <u>conjunction</u> of p and q; p and q are the <u>conjuncts</u> of the conjunction.

<u>Disjunction</u>.

Alternation or disjunction is the first of the operations for which the departure from English usage requires comment. '$p \vee q$' is true if p is true, or if q is true, <u>or</u> if both p and q are true (the <u>inclusive or</u>). The truth-table is thus:

p	0	1	0	1
q	0	0	1	1
$(p \vee q)$	0	1	1	1

Frequently in English, a disjunction using <u>or</u> is intended to exclude the case in which both disjuncts are true. To express that case in the notation of the propositional calculus, it is necessary to write a more complex statement:

$$(p \vee q) \cdot (-(p \wedge q)) \qquad \text{for example}$$

that is, to exclude specifically the case in which both p and q are true. The notation '$\vee$' for alternation 'vel' as opposed to 'aut' (the <u>exclusive or</u>). p and q are the <u>disjuncts</u> of the <u>disjunction</u> $(p \vee q)$.

Comment on the form of the truth-table.

It is more usual to write truth-tables using columns rather than rows for the values of the propositional variables. The truth-table for $(p \vee q)$ is thus usually written

p	q	$(p \vee q)$
T	T	T
F	T	T
T	F	T
F	F	F

The two forms are, of course, equivalent and either form is acceptable. The one I am using makes somewhat more transparent the isomorphism with Boolean algebra, and in addition, seems easier to use.

Conditional.

The conditional $p \supset q$ read *if p then q* might be said to depart even farther from common nonmathematical usage, since it is defined by the table:

p	0	1	0	1
q	0	0	1	1
$p \supset q$	1	0	1	1

It is true: if p is false, or if q is true. It seems clear that 'if p then q' should be true if both are true, and should be false if p is true and q is false. The case, p false, q true, must be true in order that $p \wedge q \supset q$ always be true, regardless of the truth-value of p. Besides, otherwise the value is independent of the value of p, which is very uninteresting. The remaining case in which $p \supset q$ is held to be true when the antecedent p and the consequent q are both false, can be argued on the basis that an even integer should not be taken as a counterexample to if x is odd, then x^2 is odd, nor an occasional absence of smoke as a denial of the statement, if there is smoke, there is fire. The conditional thus defined is called the material conditional, to distinguish it from other possible conditional relationships, as, for example, that of cause and effect.

Biconditional.

The final common binary connective to be defined is the biconditional 'if and only if', $p \equiv q$. Its truth-table is

p	0	1	0	1
q	0	0	1	1
$p \equiv q$	1	0	0	1

$p \equiv q$ is true if p and q have the same truth-value, otherwise it is false.

Nonconjunction, nondisjunction.

There are two binary connectives which are only occasionally used, but which are interesting because each alone suffices, by compound use, to express all of the connectives given above. These are nonconjunction (Sheffer stroke) and nondisjunction (joint denial).

p	0	1	0	1
q	0	0	1	1
$(p \mid q)$	1	1	1	0
$(p \downarrow q)$	1	0	0	0

Exercise

Work out negation and conjunction for Sheffer stroke. Play with relations between connectives.

Application of propositional calculus to arguments in natural language.

The original application, or at any rate one early application, of the propositional calculus was in treating arguments of the following type:

If Jones is a communist, Jones is an atheist.
Jones is an atheist.
$\therefore$ Jones is a communist.

Let p be Jones is a communist. Let q be Jones is an atheist. Then the premises of the argument are

$$p \supset q$$

and

$$q ,$$

and the conclusion is

$$p .$$

But $(p \supset q) \wedge q$ does not logically imply p. Therefore the argument is invalid. (Work out with truth-tables.)

For some good examples of applications to mathematics, see Rosser, Logic for Mathematicians, McGraw-Hill, 1953.

Arguments can be given to show that the indiscriminate application of the propositional calculus can lead to absurdities in philosophic arguments. Professor Stevenson of Harvard gave a presentation to the philosophy club there in which he attempted to show that logic was not a fit subject for teaching to undergraduates. He pointed out, for example, that the compound statement, "If I pound on this desk at 11 o'clock, Widener Library will fall down," can be proved to be valid, since its antecedent is false. Consider also the following discussion. A If Reagan is elected, California will be a better place to live. B. That's false. A. You have just asserted that Reagan will be elected. Or "If it rains, I wear a raincoat" hence "If I don't wear a raincoat, it doesn't rain." Such matters will not concern us here.

Figure 1

negation	not p	-p ~p		p	0	1				7 is a prime.
		$\bar{p}$ ¬p		¬p	1	0				7 is not a prime.
conjunction	p and q	(p ∧ q)	conjuncts	p	0	1	0	1		7 is a prime.
		pq		q	0	0	1	1		11 is an even number.
		p • q		(p ∧ q)	0	0	0	1		
		p & q								
disjunction	p or q	(p ∨ q)	disjuncts	(p ∨ q)	0	1	1	1		
conditional	if p then q	(p ⊃ q)	antecedent consequent		1	0	1	1		if 7 is a prime, then 11 is an odd number.
biconditional	p iff q	(p ≡ q)			1	0	0	1		
nonconjunction	not both p and q	p | q	Sheffer stroke		1	1	1	0		
nondisjunction	neither p nor q	p ↓ q	joint denial		1	0	0	0		

DEFINITION: The symbols $\sim$, $\wedge$, $\vee$, $\supset$, and $\equiv$ (we exclude now $|$ and $\downarrow$) will be called <u>propositional</u> connectives.

DEFINITION: (informal) A <u>statement form</u> of the propositional calculus is an expression built up from the <u>statement letters</u> $p, q, r, p_1, \ldots$ by appropriate application of the propositional connectives.

<u>Notation</u>: We use $A, B, \ldots$ as variables over statement forms and $p, q, \ldots$ as statement letters (Mendelson uses $\mathfrak{A}, \mathfrak{B}$, for variables over statement forms, and $p, q, \ldots$ for letters.)

DEFINITION. <u>statement form</u>

1. Any statement letter is a statement form.
2. If A and B are statement forms, so are $(-A)$, $(A \wedge B)$, $(A \vee B)$, $(A \supset B)$, and $(A \equiv B)$.
3. Extremal clause

Comments on extremal clause:

1. Only those expressions are statement forms which are determined to be so by means of (1) and (2).
2. C is a statement form if and only if there is a finite sequence $A_1, A_2, \ldots, A_n$ $(n \geq 1)$ such that $A_n = C$, and if $1 \leq i \leq n$, A_i is either

a statement letter or is a negation, conjunction, disjunction, conditional or biconditional constructed from previous expressions in the sequence.

3. An expression is a statement form if and only if it can be shown to be a statement form on the basis of clauses (1) and (2).

4. The only statement forms are those given by (1) and (2).

5. An expression is a statement form if and only if it is so by virtue of (1) and (2).

<u>Note</u>: Excludes $(A \downarrow B)$.

Also excludes infinite case, $((A_1 \vee A_2) \vee A_3) \ldots$.

<u>Parentheses</u>

Note that under this definition $A \vee B$ is not a statement form because there are no parentheses.

While a statement form must, by definition, have parentheses associated with each of the connectives, conventions are usually made about abbreviated forms with fewer parentheses. If the parentheses are omitted, according to some rule, the expression is treated as if it were the statement form of which it is an abbreviation.

Standard conventions for the restoration of omitted parentheses are the following:

1. Outer parentheses are omitted.

2. Associate from the left for any one connective.

3. The connectives are ordered: $\sim$, $\wedge$, $\vee$, $\supset$, $\equiv$.

From L to R they each apply to the smallest possible scope. $p \vee -q \supset r \equiv p$ thus abbreviates $(((p \vee (-q)) \supset r) \equiv p)$

<u>Dot notations</u>

In addition to the conventions about omitted parentheses, there are several dot notations in use. These tend to strengthen the associated connectives, that is, to move them to the right in the ordering given. Whitehead and Russell (Principia Mathematica), Church, and Quine all have slightly different conventions. Examples:

PM *3.3 $\quad \vdash ((((p \supset q) \supset r) \supset (p \supset (q \supset r)))$

$$\vdash : p : q : \supset : r :: \supset :: p : \supset : q \supset r$$

Church $\quad p \supset (q \supset r) \supset . (p \supset q) \supset (p \supset r)$

In general, it seems best in an informal treatment to avoid the use of dots by the use of parentheses; however, one should be aware that these conventions exist and that they differ from one another slightly.

Evaluation of a Statement Form

To far we have given truth-table definitions for the connectives which have given us a means of evaluating, i.e., finding the truth-value of, any expression with one connective. This method can be extended step by step to obtain an evaluation for any statement form, since the form is built up by individual applications of the connectives.

Thus, for every assignment of truth-values to the statement letters of a statement form, there corresponds a truth-value for the statement form.

Example:

$$(p \supset (q \supset r)) \supset ((p \supset q) \supset (q \supset r))$$

Thus, each statement form determines a truth-function (a function from truth-values to truth-values) (f: $(0, 1)^n \rightarrow (0, 1)$), represented by the truth-table. For n distinct statement letters, there are 2^n assignments of truth-values to the letters (columns), and thus 2^{2^n} truth-functions.

Formats for truth-tables

p	0101	0101
q	0011	0011
r	0000	1111
$\neg p$	1010	1010
$\neg p \vee q$	1011	1011
$(\neg p \vee q) \supset r$	0100	1111

usual form: $p \vee q \equiv q \vee p$

p	q	$p \vee q$	$q \vee p$	$p \vee q \equiv q \vee p$
F	F	F	F	T
T	F	T	T	T
F	T	T	T	T
T	T	T	T	T

Quine's format: $(\sim p \vee q) \supset r$

F	T	T	T	T	T
T	F	T	T	T	T
F	T	F	F	T	T
T	F	T	F	T	T
F	T	T	T	F	F
T	F	T	T	F	F
F	T	F	F	T	F
T	F	T	F	F	F

<u>Tautologies</u> (Wittgenstein)

DEFINITION: A statement form which is <u>always</u> <u>true</u>, regard- of the truth-values of its statement letter, is called a <u>tautology</u>.

(In the truth-table of a tautology, the bottom row contains only 1's .)

Example (axiom 3):

$$((\sim q \supset \sim p) \supset ((\sim q \supset p) \supset q))$$

DEFINITION: If $(A \supset B)$ is a tautology, then A <u>logically</u> <u>implies</u> B .

If $(A \equiv B)$ is a tautology, then A <u>is logically equivalent</u> to B.

[Note that by reading the horseshoe as "if... then" and the symbol '$\equiv$' as 'if and only if' we have reserved the words <u>implies</u> and <u>equivalent</u> for statements in the meta language.]

<u>Examples of tautologies</u>

$$p \vee \sim p$$

$$p \equiv \sim\sim p$$

$p \wedge q$ logically implies p.

$p \wedge (p \supset q)$ logically implies q.

$p \supset q$ and $\sim p \vee q$ are logically equivalent.

DEFINITION: A is a <u>contradiction</u> if A is false for all possible truth-value assignments to its statement letters.

DEFINITION. A is <u>satisfiable</u> if A is true for some truth-value assignment.

From the definition of tautology, it is immediately clear that the truth-tables provide an <u>effective method</u> for deciding for any given statement form, whether or not it is a tautology.

We now prove some theorems about tautologies.

THEOREM 1.1: If A and $(A \supset B)$ are tautologies, so is B.

PROOF:

Suppose there is some assignment of truth-value to the statement letters of B which makes it false. Then there is an assignment to the letters of A and B which makes B false and A true. (Since every assignment makes A true.) But then this assignment makes $(A \supset B)$ false. But this is impossible because $(A \supset B)$ is a tautology.

THEOREM 1.2: (Substitution in a tautology yields a tautology.)

If A is a tautology containing the statement letters $p_1, p_2, \ldots, p_n$ and B arises from A by substitution of the statement forms $A_1, \ldots, A_n$ for $p_1, p_2, \ldots, p_n$ throughout, then B is a tautology.

PROOF:

Consider any assignment to the statement letters of $A_1, \ldots, A_n$. It gives an assignment of truth-values to $A_1, \ldots, A_n$ say $x_1, \ldots, x_n$. Then the truth-value of B is the same as the value of A under the assignment of the x_i to the p_i. But since A is a tautology, this value is T. But this was true for any assignment to the statement letters of B. Hence B is a tautology.

REMARKS:

However, if we begin with a statement form which is not a tautology, we can, by substitution, obtain a tautology. This is true with only one exception (that is, except when the statement form is a contradiction).

THEOREM 1 3: (Equivalence Theorem)

If C' arises from C by substitution of B for one or more occurrences of A, then

$$(1) \qquad ((A \equiv B) \supset (C \equiv C'))$$

is a tautology. Hence, if A is logically equivalent to B, then C' is logically equivalent to C.

PROOF:

Consider any assignment of truth-values to the statement letters of (1). If under the assignment A and B have different truth-values, then (1) is true, by the truth-table for the conditional. If they have the same truth-values, then C and C' will have the same truth-values.

The final statement follows, by the definition of logically equivalent, and Theorem 1 1.

Example 1.

Would you believe $pq \vee \overline{pr} \vee qrs \equiv pq \vee \overline{pr}$? But $pq \vee \overline{pr} \vee qrs \leftrightarrow pq \vee \overline{pr} \vee (qrs \wedge (p \vee \overline{p}))$ [where '$\leftrightarrow$' means, temporarily, has same value for all truth assignments], because $p \vee \overline{p} \leftrightarrow 1$ and $A \wedge 1 \leftrightarrow A$.

$\leftrightarrow pq \vee \overline{pr} \vee pqrs \vee \overline{p}qrs$ because $B \wedge (C \vee D) \leftrightarrow (C \wedge B) \vee (D \wedge B)$

$\leftrightarrow pq \vee pqrs \vee \overline{pr} \vee \overline{p}rqs$ rearranging.

$\leftrightarrow pq \vee \overline{p}r$ because $A \vee (A \wedge B) \leftrightarrow A$.

Example 2.

$$(p \supset q) \supset ((r \supset q) \supset (p \vee r \supset q))$$

$\leftrightarrow \overline{\overline{p} \vee q} \vee (\overline{\overline{r} \vee q} \vee (\overline{p \vee r} \vee q))$

by $A \supset B \leftrightarrow \sim A \vee B$

$\leftrightarrow (\overline{\overline{p}} \wedge q) \vee (\overline{\overline{r}} \wedge \overline{q}) \vee (\overline{p \vee r}) \vee q$

$\overline{A \vee B} \leftrightarrow \overline{A} \wedge \overline{B}$

$\leftrightarrow (p \wedge \overline{q}) \vee (r \wedge \overline{q}) \vee (\overline{p \vee r}) \vee q$

$\overline{\overline{B}} \leftrightarrow B$

$\leftrightarrow ((p \vee r) \wedge \overline{q}) \vee q \vee \overline{p \vee r}$

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

$\leftrightarrow (((p \vee r) \vee q) \wedge (\bar{q} \vee q)) \vee \overline{p \vee r}$

$\leftrightarrow (p \vee r) \vee q \vee \overline{p \vee r}$

$\leftrightarrow 1$

What is a truth assignment? Usually say a truth assignment $\mathcal{T}$ to the letters of a formula A, i.e., if A has a statement letters $(p_1, \ldots, p_n)$ each p_j is replaced by 0 or 1. Will write $\mathcal{T}A = 0$ or 1 as value of A under truth assignment $\mathcal{T}$.

A truth function f is a mapping $f\colon (0, 1)^n \to (0, 1)$.

Every statement form of n letters generates by its truth-table an n-ary truth function, obviously.

THEOREM 1.4:

Every truth function of n variables is generated by some statement form with n statement letters.

PROOF: (by construction)

Let $f(x_1, \ldots, x_n)$ be a truth function. We can express this function by a table giving the value of the function as the last line.

	x_1	0101	...	0101
	x_2	0011	...	0011
	x_3	0000		1111
		...		...
	x_n	0000		1111
e.g.	f	1011		0010

There are 2^n columns, n rows. (Explain.)

For $1 < i < 2^n$, let $C_i = \wedge_{j=1}^{n} U_j$ where U_j is P_j or $\sim P_j$ according as the entry in j^{th} row, i^{th} column is 1 or 0. Let $D = \vee C_k$ where k ranges over only those columns in which f is true. Then f is truth function corresponding to D. For, if $\mathcal{T}$ is any assignment to $(p_1, \ldots, p_n)$ then there is a corresponding column k of the above table such that $\mathcal{T}C_k = 1$ and $\mathcal{T}C_i = 0$ $(i \neq k)$. If f is true at $\mathcal{T}$ then row f, column k is 1; so C_k is a disjunct of D; so $\mathcal{T}D = 1$. If f is false at $\mathcal{T}$, then row f, column k, is 0; so C_k isn't disjunct of D; so $\mathcal{T}D = 0$.

This completes the proof except when the truth function is identically false. The construction then produces nothing. Take D as $p_1 \wedge \bar{p}_1$.

Example:

x_1	0101
x_2	0011
f	1101

$$D = \bar{p}_1\bar{p}_2 \vee p_1\bar{p}_2 \vee p_1p_2$$

BLANK PAGE

NORMAL FORMS IN THE PROPOSITIONAL CALCULUS

DEFINITION: A <u>literal</u> is a statement letter or the negation of a statement letter.

Notation

$\hat{p}$ and $\overset{\vee}{p}$ are used as variables over the signed statement letters p and $\bar{p}$. If $\hat{p}$ is p, then $\overset{\vee}{p}$ is $\bar{p}$. If $\hat{p}$ is $\bar{p}$, then $\overset{\vee}{p}$ is p.

DEFINITION: A statement form is in <u>disjunctive</u> (<u>conjunctive</u>) <u>normal form</u> if it is a disjunction (conjunction) consisting of one or more disjuncts (conjuncts) each of which is a conjunction (disjunction) of one or more literals (abbreviated <u>d.n.f.</u>, <u>c.n.f.</u>).

DEFINITION: In speaking of a d.n.f. (c.n.f.) we refer to the disjuncts (conjuncts) as <u>clauses</u>.

THEOREM 1.5: Every statement form is logically equivalent to a statement form in d.n.f.

Every statement form is logically equivalent to one in c.n.f.

PROOF:

<u>For the d.n.f.</u>: Corollary to the proof of Theorem 1.4. That is,

(1) any contradiction is logically equivalent to $p \wedge \bar{p}$,

(2) if it is not a contradiction, then its truth-table has at least one 1 . The alternation of the C_k corresponding to the 1's in the truth-table is equivalent to the original form, and is in disjunctive normal form.

<u>For the c.n.f.</u>: The d.n.f. of

$$
\begin{aligned}
-A \text{ is } & A_1 \vee A_2 \vee \ldots \vee A_n \\
-A \text{ eqv. } & A_1 \vee A_2 \vee \ldots \vee A_n \\
A \text{ eqv. } & \sim(A_1 \vee \ldots \vee A_n) \\
\text{eqv. } & \sim A_1 \wedge \sim A_2 \wedge \ldots \wedge \sim A_n \\
\text{eqv. } & B_1 \wedge B_2 \wedge \ldots \wedge B_n \\
\text{eqv. } & \text{c.n.f.}
\end{aligned}
$$

DEFINITION: The <u>full disjunctive normal form</u> (f.d.n.f.) of a statement form A is a logically equivalent statement form which is in d.n.f. and in which

1. in each clause every letter of A occurs exactly once; and
2. no two clauses contain precisely the same literals (no duplicates).

THEOREM 1.6: Every non-contradictory (non-tautologous) form has a f.d.n.f. (f.c.n.f.) which is unique to within order.

PROOF:

The construction for Theorem 1.5 in fact produced a f.d.n.f. and f.c.n.f. It is unique to within order since any form having different clauses will have different truth-tables.

<u>Algorithm for Obtaining Disjunctive Normal Form</u>

1. Eliminate unwanted connectives.
2. Push negation all the way in.
3. Multiply out the conjunctions.

METHODS for obtaining f.d.n.f.:

1. The truth-table method given by the proof of the theorem.
2. Suppose A is any non-contraction. Put into d.n.f. using equivalences. Then if any clause A is missing a letter, say p, replace A_i by $(p \vee \bar{p}) \& A_i$. This becomes $p \& A_i \vee \bar{p} \& A_i$. Eliminate duplicates and any $p\bar{p}$'s and repeat until f.d.n.f. is obtained.

Methods for obtaining f.c.n.f. are analogous (dual).

Examples:

$$p \supset q$$

d.n.f. $\bar{p} \vee q$

$$\bar{p}q \vee \bar{p}\bar{q} \vee pq \vee \bar{p}q$$

f.d.n.f. $pq \vee \bar{p}\bar{q} \vee pq$

$(p\bar{q} \vee \bar{p}q)$

$(p \vee \bar{p}q)(\bar{q} \vee \bar{p}q)$

$(p \vee \bar{p})(p \vee q)(\bar{q} \vee \bar{p})(\bar{q} \vee q)$

$(p \vee \bar{q})(\bar{q} \vee \bar{p})$

$(p \vee q)(\bar{p} \vee \bar{q})$

$(\bar{p} \vee r)(p \vee r)$

$pr \vee rp \vee rr$

$\bar{p}r \vee rp$

not just r

$p \equiv \neg p$

a contradiction.

No f.d.n.f.; f.c.n.f. $p \wedge \bar{p}$ okay.

COROLLARY 1.7: An f.d.n.f. with n letters is a tautology if and only if it has 2^n clauses.

PROOFS:

1. By the truth-table argument.
2. Suppose the clause $\bar{p}q\bar{r}$ is missing. Then the truth-value assignment 010 will make A false. On the other hand, if all clauses appear, there is one which is true for any assignment.
3. By factoring, by the use of equivalences and the distributive law, we can reduce to $p \vee \bar{p}$ which is a tautology.

c.n.f. taut.

THEOREM 1.8: A necessary and sufficient condition that a form A be a tautology is that in every clause of the c.n.f. at least one letter appears both negated and unnegated.

PROOF: Assume A is a tautology. Let A' be a c.n.f. of A . It is identically true. Hence every clause must be identically true. But a clause A_i is an alternation of literals and hence can be identically true iff some one letter occurs both negated and unnegated.

d.n.f. contra-diction

THEOREM 1.9: Dual statement for d.n.f.: A n.s.c. that a d.n.f. be a contradiction is that in every clause some letter occurs both negated and unnegated.

PROOF: Dual to the above.

BLANK PAGE

Decision methods in the Propositional Calculus

We have shown that for any statement form of the propositional calculus we can test whether or not it is always true, (identically true), i.e., whether or not it is a tautology. That is to say, we have a decision procedure for the propositional calculus The decision procedure is effective and general.

By effective we mean, roughly, that there is a purely mechanical way of carrying out the method, which does not require the exercise of ingenuity. (Church)

By general, we mean that the method applies to every problem in the class. Note that the class of problems is infinite.

Restated: The decision problem for the propositional calculus is the problem of deciding effectively for any given statement form, whether or not it is a tautology.

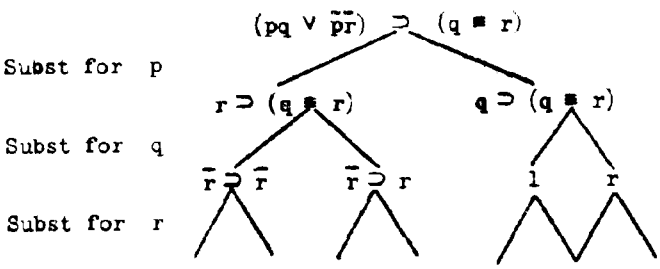
Decision methods

1. Truth-table.
2. Put in f.d n.f -- if 2^n clauses, it is a tautology by a theorem above
3. Quines (resolution) method. Form a tree, substitute at each level 0 or 1 for one letter. As the substitutions are made, evaluate by the following rules.

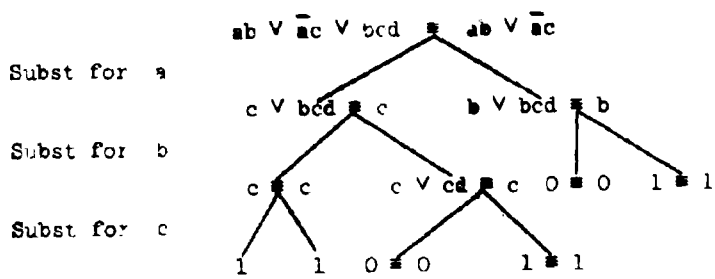
$\sim 0 \leftrightarrow 1$	$0 \wedge A \leftrightarrow 0$	$\equiv C$
$\sim 1 \leftrightarrow 0$	$1 \wedge A \leftrightarrow 1$	$0 \equiv A \leftrightarrow \sim A$
$0 \vee A \leftrightarrow A$	$0 \supset A \leftrightarrow 1$	$1 \equiv A \leftrightarrow A$
$1 \vee A \leftrightarrow 1$	$A \supset 0 \leftrightarrow \sim A$	

Continue until either some branch comes to 0 --not tautology, or all [illegible] to [illegible] tautology

<u>Example 1</u>



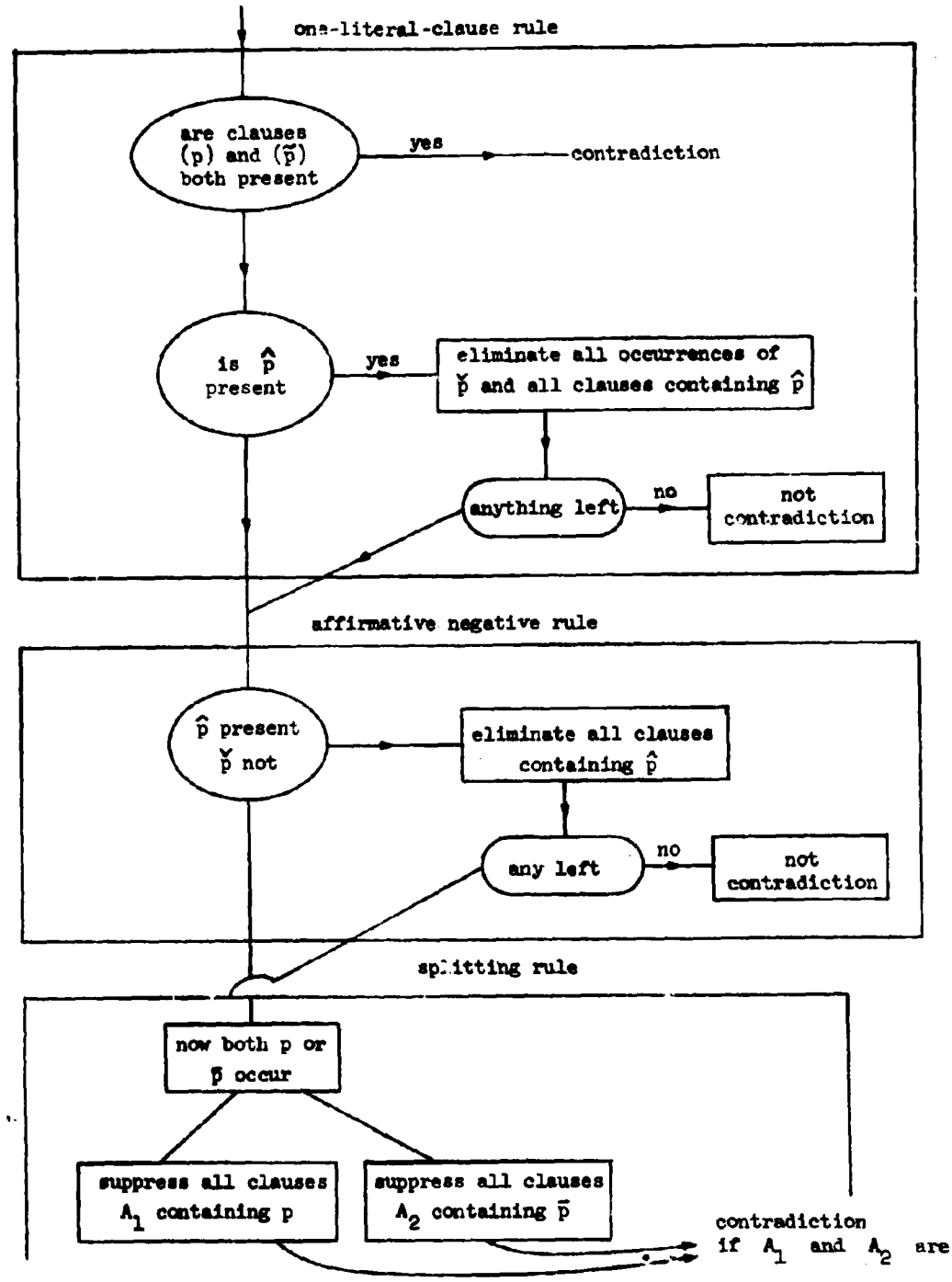
<u>Example 2</u>



<u>Davis-Putnam algorithm</u>

Tests a c n.f. for contradiction.

Step 0: May assume no clause contains p and $\bar{p}$. Any such clause can be removed (if all, then not contradiction).


one-literal-clause rule
are clauses (p) and ($\bar{p}$) both present
yes
contradiction
is $\hat{p}$ present
yes
eliminate all occurrences of $\check{p}$ and all clauses containing $\hat{p}$
anything left
no
not contradiction
affirmative negative rule
$\hat{p}$ present $\check{p}$ not
eliminate all clauses containing $\hat{p}$
any left
no
not contradiction
splitting rule
now both p or $\bar{p}$ occur
suppress all clauses A_1 containing p
suppress all clauses A_2 containing $\bar{p}$
contradiction if A_1 and A_2 are

Example 1.

$$(\bar{p},\ q)(\bar{q},\ r)(p)(\bar{r})$$

which is:

$$\sim((p \supset q) \ \wedge \ (q \supset r) \ \supset \ (p \supset r))$$

by OLCR:

$$(\bar{p},\ q)(\bar{q})(p)$$

$$(q)(\bar{q})$$

★ contradiction

Example 2.

$$(p,\ q)(p,\ \bar{q})(\bar{p},\ q)(\bar{p},\ \bar{q})$$

$$(q)(\bar{q}) \qquad\qquad (q\ \bar{q})$$

both contradictions

THEOREM 1.10: Davis-Putnam procedure works.

PROOF:

Note that there is a dual procedure for testing if d.n.f is a tautology.

Dunham's Elimination Theorem

THEOREM: Let A be the disjunctive formula

$$\hat{p}A_1 \vee \ldots \vee \hat{p}A_j \vee \ldots \vee \hat{p}A_n \vee \overset{\vee}{p}B_1 \vee \ldots \vee \overset{\vee}{p}B_m \vee C$$

where the letter p does not occur in A_j $(j = 1, \ldots, n)$, B_ℓ $(\ell = 1, \ldots, m)$ or C, and where, for all ℓ $A_1 B_\ell$ is a contradiction. Let A' be

$$\hat{p}A_1 \vee \ldots \vee \hat{p}A_{j-1} \vee \hat{p}A_{j+1} \vee \ldots \vee \hat{p}A_n \vee \overset{\vee}{p}B_1 \vee \ldots \vee \overset{\vee}{p}B_m \vee C$$

Then A is a tautology if and only if A' is a tautology.

PROOF:

Every clause of A' is also a clause of A, so if A' is a tautology, so is A. Thus it is only necessary to prove that if A is a tautology, then so is A'. We prove this by showing that there is no assignment of truth-values to the letters of A which makes A true and A' false.

Case 1. $\hat{p}$ is p.

Then

$$A \text{ eqv. } p(A_1 \vee \ldots \vee A_j \vee \ldots \vee A_n) \vee \bar{p}(B_1 \vee B_2 \vee \ldots \vee B_m \vee C$$

$$A' \text{ eqv. } p(A_1 \vee \ldots \vee A_{j-1} \vee A_{j+1} \vee \ldots \vee A_n) \vee \bar{p}(B_1 \vee B_2 \vee \ldots \vee B_m) \vee C$$

Let all the letters of A be $(p, p_1, \ldots, p_k)$ and suppose the truth-value assignment $(a, a_1, \ldots, a_k)$ makes A true, A' false.

Case 1a <u>a is 0</u>.

Then $(0, a_1, \ldots, a_k)$ makes A true, A' false, hence $(B_1 \vee \ldots \vee B_m) \vee C$ false. This is absurd.

Case 1b. <u>a is 1</u>.

Then $(1, a_1, \ldots, a_k)$ makes A true, A' false, hence $(A_1 \vee \ldots \vee A_n \vee C$ true and $(A_1 \vee \ldots \vee A_{i-1} \vee A_{i+1} \vee \ldots \vee A_n) \vee C$ false. Hence it must make A_i true and C false. But then, since A_i and C do not contain p, $(0, a_1 \ldots a_k)$ also makes A_i true and C false. But since A is a tautology $(0, a_1, \ldots, a_k)$ must make $(B_1 \vee \ldots \vee B_m)$ true. But it is not possible to make both A_i and $(B_1 \vee \ldots \vee B_m)$ true, since to do so would make A_iB_j true for at least one j, contrary to hypothesis. Hence this case is also impossible.

Case 2 <u>$\hat{p}$ is p</u>

By symmetry.

<u>DUNHAM'S METHOD</u>

For each letter p in the d.n.f. formula A, circle the occurrence of $\hat{p}$ in clause A_i iff A also contains a

clause $\overset{\vee}{p}B_j$ such that A_iB_j is not a contradiction. Delete all clauses that contain uncircled literals. Erase circles and repeat until at some step no clause is deleted. If there are no clauses left, A is not a tautology. If the d.n.f. formula A' remains, then A is a tautology iff A' is a tautology.

<u>REMARK</u>: To contrast with other methods, take $pq \vee \bar{p}q \vee p\bar{q}$

BLANK PAGE

AN AXIOM SYSTEM FOR THE PROPOSITIONAL CALCULUS

NOTA BENE: These lectures are intended as comments on Section 4, Chapter 1 of Mendelson. They are in no sense complete, but are intended to assist in reading the text. They do not replace the text, which is essential.

Reasons for wanting to construct a formal system:

1. To be used later in quantification theory.
2. There are interesting subsystems of the propositional calculus.
3. For a simple illustration as an introduction to the basic notions of formal systems.

Formal Theory

1. Countable set of symbols. (Normally constructed from a finite set of symbols.)
2. Well-formed formulae (wff's). This must be effective.
3. Axioms. If effective then an axiomatic theory. Example of a non-axiomatic theory would be to take as axioms the theorems of the first-order predicate calculus.
4. Rules of inference. Again effectively decidable.

Proof

Theorem

Decidable vs. undecidable theories

The formal axiomatic theory L for the propositional calculus

primitive symbols

wffs

If A, B, and C are any wffs of L, then the following are axioms of L

A1 $(A \supset (B \supset A))$

A2. $((A \supset (B \supset C)) \supset ((A \supset B) \supset (A \supset C))$

A3. $((\sim B \supset \sim A) \supset ((\sim B \supset A) \supset B))$

Remarks: Negation occurs only in A3. The system with axioms A1 and A2 is called the positive implicational calculus and is decidable (Arnold Schmidt). schema, schemata (schemas). We omit parens. as abbreviation.

Rule of inference MP. Note that with more rules of inference we could have fewer axioms. In particular, with a substitution rule we could have a finite axiom set.

Prove that the set of axioms is effective, i.e., L is an axiomatic theory.

Note now that the axioms are all tautologies.

To show that L is in fact the system we want, we will prove the following metatheorems. (def.)

1. Soundness. Every theorem is a tautology.

(Verify that so far okay, all axioms are tautologies.)

2. <u>Completeness</u>. Every tautology is a theorem.

3. <u>Consistency</u>. For no wf A, both A and ~A are theorems of L .

<u>Absolute consistency</u>. Some wf A is not a theorem of L . (For if the system does not have negation, we cannot prove consistency as defined above.)

<u>Absolute completeness</u> (supersaturation). If we add another schema, which is obtained from a statement form that is not a theorem, the result is inconsistent. (Exercise.)

Prior to proving the main metatheorems we will want some theorems.

LEMMA 1.7:

Heuristic argument.

This is a proof schema, not a proof.

Note that the comments on the right are not part of the proof.

They can be effectively recovered from the proof.

Note that A3 is not used.

<u>DEDUCTION AND THE DEDUCTION THEOREM</u>

Definition of deduction.

Three properties of the notion of consequence.

Property (1) does not always hold...in particular if there is a substitution rule of inference.

Note that not every line of a deduction is a tautology.

Deduction theorem Herbrand.

Examples of usefulness. Prove corollary 1.9ii. Lemma 1.10a

Proof of the deduction theorem.

Remark only axioms A1 and A2 are used in proof plus MP.

constructive

Note LEMMA 1.10 -- It is essential to the proofs of the main metatheorems

PROP 1.12 <u>Every theorem is a tautology</u> soundness

Proof by induction on the lines of a proof.

PROP 1.13 <u>Every tautology is a theorem of L</u>. completeness.

Needs:

LEMMA 1.12

Note that the object is not to show that A' is true under the hypotheses, but that A' is provable from the hypotheses

Let A be a wff and let $p_1, \ldots, p_k$ be the statement letters occurring in A. For a given assignment of truth-values to $p_1, \ldots, p_k$, let p_i' be p_i if p_i takes the value T; and let p_i' be $\sim p_i$ if p_i takes the value F. Let A' be A, if A takes the value T under the assignment; let A' be $\sim A$ if A takes the value F. Then $p_1', \ldots, p_k' \vdash A'$.

PROOF

Alternative Axiomatizations of the Propositional Calculus

P_1: $p \supset (q \supset p)$

$p \supset (q \supset r)) \supset ((p \supset q) \supset (p \supset r))$

$(((q \supset f) \supset f) \supset p)$

or: $(\sim p \supset \sim q) \supset (q \supset p)$ [This is Lemma 1.10d Exercise 42.1]

Rules of inference

MP

SUBST: From A to infer $S^b_B A|$.

Note now the necessary redefinition of a deduction.

Each line is: a hypothesis

or a variant of an axiom

or is inferred by MP from two preceding lines

or is inferred by SUBST from a preceding line, where the variable substituted <u>for</u> does not occur in the hypotheses.

Example of violation:

$p \vdash p$ (hypothesis)

$p \vdash \sim p$ (subst)

$p \supset \sim p$ deduction theorem

All tautologies are axioms.

Historical Notes

Propositional Calculus

Quine, Preface to Methods of Logic, "Logic is an old subject, and since 1879 it has been a great one."

Frege Begriffsschrift 1879

P_F: $p \supset (q \supset p)$

$p \supset (q \supset r) \supset. (p \supset q) \supset (p \supset r)$

$(p \supset (q \supset r)) \supset. q \supset (p \supset r)$

$(p \supset q) \supset (\sim q \supset \sim p)$

$\sim\sim p \supset p$

$p \supset \sim\sim p$

Rules of inference are MP and SUBST.

Exercise: Discuss the independence of the axioms.

Frege's work was not known to Whitehead and Russell when they began. Later they were perhaps the first to appreciate its significance.

Use of axiom schemata: von Neumann 1927

Use of all tautologies as axioms: Herbrand 1930

P_1 Wajsberg 1930 has constant f

P_2 Łukasiewicz 1930 (from P_F) no constants

Frege s notation

$\vdash A$ judgement

$\vdash \begin{array}{l} A \\ \quad B \end{array}$ $B \supset A$ $\sim(B \& \sim A)$

conditional stroke

MP as only rule of inference (but in fact SUBST is needed)

from

$\vdash \begin{array}{l} A \\ \quad B \end{array}$ and $\vdash A$ to infer $\vdash B$

$\vdash A$ $\sim A$

$\vdash \begin{array}{l} A \\ \quad B \end{array}$ $B \supset \sim A$ $\sim(B \& \sim A)$ i.e., $\sim(B \& A)$

$\sim B \vee \sim A$ $\sim(B \& A)$

$\vdash \begin{array}{l} A \\ \quad B \end{array}$ $\sim B \supset A$ $B \wedge A$

$\vdash \begin{array}{l} A \\ \quad B \end{array}$ $\sim(B \supset \sim A)$ $\sim(\sim B \vee \sim A)$ $B \wedge A$

Note that he has <u>implications</u> and <u>negations</u> (only).

---

fn. argument

$\vdash\!\!-\!\!\smile^{a}\!\!-\ \Phi\ (a)$ $\qquad$ $(\forall a)\ \Phi\ (a)$ $\qquad$ "The number 20 can be represented as the sum of four squares." "Any number."

$\vdash\!\!-\!\!\smile^{a}\!\!-\ \Phi\ (a)$

$\vdash\ A$

$\quad\ \smile^{a}\!\!-\ x\ (a)$ $\qquad$ scope $\smile$

$\vdash\!\!-\!\!\smile^{a}\!\!-\ A\ (a)$

$\quad\ \smile^{e}\!\!-\ B\ (a,\ e)$

$\vdash\!\!-\!\!\smile^{a}\!\!-\ X\ (a)$ $\qquad$ <u>there are x's</u> $\qquad$ $(\exists a)X(a)$

First-order Predicate Calculus

Introduction:

In the propositional calculus we dealt with logical inferences using statement letters (which represented sentences as unbroken units) and the logical connectives. But, as we pointed out, even the simple classical syllogism was beyond the scope of that system.

In the first-order predicate calculus (or functional calculus) we deal with the internal structure of sentences, using symbols for properties or relations, and for expressions such as all, any, some.

Consider the example:

1. There is a man who is physician to everyone.
2. Everyone has some physician.

It is easy to see that logically 2 follows from 1. However, the argument lies outside the scope of the propositional calculus. It is just this sort of argument for which the predicate calculus is suited. It can be expressed as:

1. $(Ex)\ (y)\ F(x, y)$
2. $(y)\ (Ex)\ F(x, y)$

$F(x, y)$ stands for x is physician to y . The existential quantifier (Ex) means there is some x, and the universal quantifier (y) means for all y . The argument is valid, moreover, for an arbitrary binary relation F, and for an arbitrary set as the range of the individual variables x and y .

The formal system of the propositional calculus was designed to catch as theorems all the universally true statements or tautologies. For the predicate calculus we similarly wish to have as theorems all the statements which are <u>valid</u>, that is which are true in every non-empty universe. In proving completeness, we will show that all such valid wffs are theorems.

Formal development of the predicate calculus

(Note that the system to be developed here differs from Mendelson's and is essentially a subset of it. Mendelson includes <u>individual constants</u> and <u>function letters</u>.) The system we present here is the <u>pure first-order predicate calculus</u>.

Primitive symbols:

Individual variables: $x, y, z, x_1, y_1, z_1, x_2, \ldots$

Predicate letters:

Monadic: $F^1, G^1, H^1, F^1_1, G^1_1, H^1_1, F^1_2, \ldots$

Dyadic: $F^2, G^2, H^2, F^2_1, G^2_1, H^2_1, F^2_2, \ldots$

j-adic: $F^j, G^j, H^j, F^j_1, G^j_1, H^j_1, F^j_2, \ldots$

(For each j, an infinite number of j-adic predicate letters.)

Connectives: $\sim \supset$, ( )

Metalanguage: $A, B, \ldots$ for wffs.

x, y, z for variables.

Atomic formulas

If F_i^n is a predicate letter and $x_{i_1}, x_{i_2}, \ldots, x_{i_n}$ are individual variables (not necessarily distinct), then $F_i^n(x_{i_1}, \ldots, x_{i_n})$ is an atomic formula.

Convention

Superscripts on predicate letters may be omitted, since it is always clear from the context in any wff what the superscripts must be. In the formula

$$F_1(x_1, x_2) \supset F_1(x_2)$$

it is clear that the two predicate letters are in fact distinct. Although it is more usual to then use different letters:

$$F_1(x_1, x_2) \supset F_2(x_2)$$

Well-formed formulas

1. Every atomic formula is a wff .
2. If A and B are wffs, and y is a variable, then $(\sim A)$, $(A \supset B)$, and $((y)A)$ are wffs .
   Comment: Note that we do not require that y occur (free) in A .
3. Extremal clause.
   Note that the rules for a wff are effective.

Pure first-order predicate calculus

first-order: no quantification over predicate letters.

pure: no individual constants, no function symbols.

(Hence the only terms are variables.)

predicate calculus: no nonlogical axioms (as opposed to a "first-order theory").

Definition

In the expression $((y)A)$

(y) is a universal quantifier

A is the scope of the quantifier (y)

Alternate notation: $(\forall y)$

Conventions

Parentheses omitted as before.

The scope of a quantifier is to be taken as small as possible. In $(y)A \vee B$ the scope of the quantifier is A.

Definition:	$(\exists x_i)\ A$	stands for $\sim(x_i) \sim A$
	$(\exists x_i)$ or (Ex_i)	is an existential quantifier
	$A \vee B$	stands for $(\sim A) \supset B$
	$A \wedge B$	stands for $\sim(A \supset \sim B)$
	$A \equiv B$	for $(A \supset B) \wedge (B \supset A)$

Free and bound variables

Definition: A particular occurrence of the individual variable x_i in A is a bound occurrence if x_i is the variable of the quantifier (x_i) or if it is within the scope of a quantifier (x_i). Otherwise, the occurrence is free.

$$(x_1)F_1(x_1, x_2) \qquad F_2(x_1)$$

Definition: A variable is bound in A if it has at least one bound occurrence in A. Similarly, free in A.

Remark: A variable may be both bound and free in A.

Definition: If A is a wff and x_j and x_i are variables, then x_j is said to be free for x_i in A iff no free occurrences of x_i lie within the scope of any quantifier (x_j).

Informally: If we substitute x_j for free x_i throughout A, then x_j is never captured by a quantifier (x_j). Note also that (Ex_j) is $\sim(x_j)\sim$.

Substitution convention

$A(x_{i_1}, \ldots, x_{i_n})$ is used to denote a wff which may have some of $x_{i_1}, \ldots, x_{i_n}$ free. Then $A(x_{j_1}, \ldots, x_{j_n})$

is the result of substituting x_j for x_i at all of the free occurrences of x_i .

Examples:

If $A(x_1, x_2)$ is $A_1(x_2) \vee A_2(x_1)$

then $A(x_1, x_1)$ is $A_1(x_1) \vee A_2(x_1)$

and $A(x_3, x_1)$ is $A_1(x_1) \vee A_2(x_3)$.

<u>Axiom schemata</u>

If A, B, and C are wffs, and x and y are variables, then the following are axioms:

1. $A \supset (B \supset A)$
2. $(A \supset (B \supset C)) \supset ((A \supset B) \supset (A \supset C))$
3. $(\sim B \supset \sim A) \supset ((\sim B \supset A) \supset B)$
4. $(x)A(x) \supset A(y)$

   <u>if</u> y is free for x in A(x) .
5. $(x)(A \supset B) \supset (A \supset (x)B)$

   <u>if</u> A contains no free occurrences of x .

Remarks: $(x)A \supset A$ is a special case of Axiom 4. Verify validity.

<u>Rules of inference</u>

1. <u>Modus ponens</u>. (MP) From A and $(A \supset B)$ to infer B .

   (i.e., B is a direct consequence of A and $A \supset B$.)
2. <u>Generalization</u>. (GEN) From A to infer (x)A .

Remark: ? (GEN) cannot be rephrased as an axiom

$$A \supset (x)A$$

since this would give

$$A(x) \supset (x)A(x)$$

or

$$(x)(A(x) \supset (x)A(x))$$

i.e.,

$$(y)(A(y) \supset (x)A(x))$$

Take $A(x)$ as "x is a prime".

Remark: We say x is generalized on.

Violations for A4, A5:

$$(x)(\exists y)F(x, y) \supset (\exists y)F(y, y)$$ Take F as mother

$$(x)(F_1(x) \supset F_2(x)) \supset (F_1(x) \supset (x)F_2(x))$$ Take F_1 even, F_2 even square

o.k.:

$$(x)((\exists y)G_1(y) \supset G_2(x)) \supset ((\exists y)G_1(y) \supset (x)G_2(x))$$

BLANK PAGE

THEOREM 2.1: Every wff A which is an instance of a tautology (of the prop. calc.) is a theorem. And it may be proved using only axioms A1-3, and MP.

PROOF: A arises from a tautology W by substitution. By the completeness of L (the propositional calculus, $\vdash W$. Now modify the proof of W by making throughout the same substitutions as were used in obtaining A from W. (For statement letters which occur in the proof which do not occur in W, put arbitrary new wffs. (This is necessary because we did not include propositional variables in our formulation of the first-order predicate calculus.)) Then the result is a proof of A (because of the use of axiom schemata). It uses only A1-A3 and MP.

THEOREM: $\vdash (y) \sim A(y) \equiv \sim (Ey)A(y)$

PROOF:	$p \equiv \sim\sim p$	Tautology (by completeness of L)
	$\vdash (y) \sim A(y) \equiv \sim\sim (y) \sim A(y)$	By the theorem above.
	$\vdash (y) \sim A(y) \equiv \sim (Ey)A(y)$	By definition of (Ey) .

CONSISTENCY

THEOREM: The first-order predicate calculus is consistent, (i.e., there is no wff A such that $\vdash A$ and $\vdash \sim A$).

PROOF:

1. Define a mapping h of the set of wffs of the predicate calculus into the set of wffs of the propositional calculus:

   Let $h(A)$ be the wff obtained from A by

   i. deleting quantifiers and variables, together with associated commas and parentheses, and

   ii. replacing distinct predicate letters by distinct statement letters.

2. If $\vdash A$ of the predicate calculus, then $\vdash_L h(A)$.

   Axioms map into tautologies. A1-A3 obviously.

   A4 into $(A \supset A)$, A5 into $((A \supset B) \supset (A \supset B))$.

   MP and GEN preserve tautologies.

3. $h(\sim A)$ is $\sim h(A)$.

   $\therefore$ if both $\vdash A$ and $\vdash \sim A$ in the predicate calculus, we would have by (2), $\vdash h(A)$ and $\vdash \sim h(A)$ in L, which contradicts the consistency of L .

DEDUCTION THEOREM (see also Mendelson 2.5)

Definition: A deduction of B from a set Γ of wffs is a finite sequence of wffs $B_1, B_2, \ldots, B_m$ of which B_m is B and for each i either

1. B_i is an axiom

or 2. B_i is a member of Γ

or 3. B_i results from B_j and B_k $(j, k < i)$ by MP

or 4. B_i results from B_j $(j < i)$ by GEN subject to the restriction that no variable free in any wff in Γ is generalized upon.

Definition: If there is a deduction of B from Γ, we write $\Gamma \vdash B$.

DEDUCTION THEOREM

If $\Gamma, A \vdash B$ then $\Gamma \vdash A \supset B$.

PROOF: Consider a deduction of B from Γ, and A:

$$\begin{aligned} \Gamma, A &\vdash B_1 \\ &\vdash B_2 \\ &\quad\vdots \\ &\quad B_h \\ &\quad\vdots \\ &\vdash B \end{aligned}$$

Induction Hypothesis: $\Gamma \vdash A \supset B_i$ for all $i < h$

To show: $\Gamma \vdash A \supset B_h$.

Case I. B_h is an axiom.

$\Gamma \vdash B_h \supset (A \supset B_h)$	A1.
$\vdash B_h$	Axiom.
$\vdash A \supset B_h$	MP.

Case II_A. B_h is A .

$\Gamma \vdash A \supset A$	Instance of tautology.

Case II_B. $B_h \in \Gamma$.

$$\Gamma \vdash B_h \supset (A \supset B_h)$$

$$\vdash B_h$$

$$\vdash A \supset B_h$$

Case III. B_h arises from B_j and B_k by MP. B_k is $B_j \supset B_h$. Continued on next page.

$\Gamma \vdash A \supset (B_j \supset B_h)$	Induction hypothesis
$\Gamma \vdash A \supset B_j$	Induction hypothesis
$\Gamma \vdash A \supset (B_j \supset B_h) \supset ((A \supset B_j) \supset (A \supset B_h))$	Axiom
$\Gamma \vdash (A \supset B_j) \supset (A \supset B_h)$	MP
$\Gamma \vdash (A \supset B_h)$	MP

Case IV. B_h arises from B_k by GEN. B_h is $(x)B_k$.

$\Gamma \vdash A \supset B_k$	
$\Gamma \vdash (x)(A \supset B_k)$	By GEN (x not free in Γ)
$\Gamma \vdash (x)(A \supset B_k) \supset (A \supset (a)B_k)$	Axiom 5 (x not free in A)
$\Gamma \vdash A \supset (x)\, B_k$	MP

Therefore, if $\Gamma, A \vdash B$, then $\Gamma \vdash A \supset B$. q.e.d.

Note that while in prop. calc. we had that if $\Gamma \vdash A$ then for any set Δ, also $\Gamma, \Delta \vdash A$ this no longer follows immediately. (But can be proved.)

USEFUL THEOREM SCHEMATA

For any wffs A and B:

1. $(y) \sim A(y) \equiv \sim (Ey)A(y)$

2. $(x_1)(x_2)A \supset (x_2)(x_1)A$

3. $(x)(A \supset B) \supset ((x)A \supset (x)B)$

4. $(x)(A \supset B) \supset ((Ex)A \supset (Ex)B)$

5. $(x)(A \wedge B) \equiv (x)A \wedge (x)B$

6. $(y_1)\ldots(y_n)A \supset A$

7. If $A(x)$ and $A(y)$ are similar

   $(x)A(x) \equiv (y)A(y)$

8. If $A(x)$ and $A(y)$ are similar

   $(Ex)A(x) \equiv (Ey)A(y)$

9. If x not free in A

   $A \equiv (x)A$

10. If x not free in A

    $A \equiv (Ex)A$

*11. If x not free in A

    $(x)(A \supset B) \equiv (A \supset (x)B)$

Comments:

- Note validity
- Note movements of quantifiers in and out
- Note $\vee$, $\wedge$

*12. If x not free in A

$$(x)(B \supset A) \equiv ((Ex)B \supset A)$$

13. $$(x)(A \equiv B) \supset ((x)A \equiv (x)B)$$

*14. $$(Ex)(A \vee (x)A) \equiv (x)A$$

*15. $$(x)A \equiv \neg(Ex)\neg A$$

*16. If x not free in A

$$((x)B(x) \supset A) \equiv ((Ex)(B(x) \supset A))$$

*17. If x not free in A

$$(A \supset (Ex)B(x)) \equiv (Ex)(A \supset B(x))$$

$$((Ex)A \vee (Ex)B) \equiv (Ex)(A \vee B)$$

19. $$((x)A \vee (x)B) \supset (x)(A \vee B)$$

20. $$(Ex)(A \wedge B) \supset ((Ex)A \wedge (Ex)B)$$

The proof of most of these are left as exercises. Proof of 16, 17, 19, and 20 will follow. These schemata are used frequently and will be referred to by a circled number, viz. (16).

16. If x not free in A, $((x)B(x) \supset A) \equiv (Ex)(B(x) \supset A)$

Proof:

Left to right:	1.	$(x)B(x) \supset A$	Hyp
	2.	$\sim(Ex)(B(x) \supset A)$	Hyp
	3.	$\sim\sim(x) \sim (B(x) \supset A)$	2, Abbreviation
	4.	$(x)(B(x) \wedge \sim A)$	3, by tautologies, & replacement thm
	5.	$B(x) \wedge \sim A$	4, Axiom 4 and MP
	6.	$B(x)$	5, tautology
	7.	$(x)B(x)$	6, GEN
	8.	A	1, 8, MP
	9.	$\sim A$	5, tautology
	10.	$A \wedge \sim A$	8, 9, tautology
11.		$(x)B(x) \supset A, \sim (Ex)(B(x) \supset A) \vdash A \wedge \sim A$	1 - 10
12.		$(x)B(x) \supset A \vdash \sim (Ex)(B(x) \supset A) \supset (A \wedge \sim A)$	11, deduction thm
13.		$(x)B(x) \supset A \vdash (Ex)(B(x) \supset A)$	12, tautology
14.		$\vdash (x)B(x) \supset A \supset (Ex)(B(x) \supset A)$	13, deduction thm

Right to left:	1.	$(x)B(x)$	Hyp
	2.	$B(x)$	1, Axiom 4 and MP
	3.	$\sim A$	Hyp
	4.	$\sim(B(x) \supset A)$	2, 3 taugology
	5.	$(x) \sim (B(x) \supset A)$	GEN
6.		$(x)B(x), \sim A \vdash (x) \sim (B(x) \supset A)$	1 - 5
7.		$(x)B(x) \vdash \sim A \supset (x) \sim (B(x) \supset A)$	6, deduction thm
8.		$(x)B(x) \vdash (Ex)(B(x) \supset A) \supset A$	7, taut., def.
9.		$(x)B(x), (Ex)(B(x) \supset A) \vdash A$	8, MP
10.		$\vdash (Ex)(B(x) \supset A) \supset ((x)B(x) \supset A)$	deduction theorem
		$\therefore \vdash ((x)B(x) \supset A) \equiv (Ex)(B(x) \supset A)$	14, 10 by taut.

6. $\vdash (y_1) \ldots (y_n)A \supset A$

PROOF:

$(y_1) \ldots (y_n)A$	(1)	$(y_1) \ldots (y_n) A$	Hyp.
	(2)	$(y_2) \ldots (y_n)A$	A4 and MP
	$\vdots$		
	(n+1)	A	

$\therefore$ by deduction theorem.

Let $A(x_j)$ arise from $A(x_i)$ by substituting x_j for all <u>free</u> occurrences of x_i.

Definition: If x_i and x_j are distinct, then $A(x_i)$ and $A(x_j)$ are <u>similar</u> iff x_j is free for x_i in $A(x_i)$ and $A(x_i)$ has no free occurrences of x_j.

Intuitively:

7. If $A(x_i)$ and $A(x_j)$ are similar, then $\vdash (x_i)A(x_i) \equiv (x_j)A(x_j)$.

PROOF:

(1)	$(x_i)A(x_i) \supset A(x_j)$	A4
(2)	$(x_j)((x_i)A(x_i) \supset A(x_j))$	GEN
(3)	$(x_i)A(x_i) \supset (x_j)A(x_j)$	A5 and MP
$\vdots$		
(5)	$(x_j)A(x_j) \supset (x_i)A(x_i)$	Similarly
$\therefore$ (7)	$(x_i)A(x_i) \equiv (x_j)A(x_j)$	$A_1 \supset (A_2 \supset A_1 \wedge A_2)$ Theorem 2.1. Def. of $\equiv$. MP twice.

8. Will follow trivially from <u>equivalence theorem</u>.

<u>Conjunction Rule</u> $A, B \vdash A \wedge B$

<u>Disjunction Rule</u>

$$A \supset C,\ B \supset D,\ A \vee B \vdash C \vee D$$

Proof by 2.1.

9. If x not free in A then $\vdash A \equiv (x)A$

(1)	$A \supset A$	Theorem 2.1
(2)	$(x)(A \supset A)$	GEN
(3)	$A \supset (x)A$	A5 and MP
(4)	$(x)A \supset A$	A4
(5)	$A \equiv (x)A$	(3) (4) def. of $\equiv$, and conjunction rule.

EQUIVALENCE THEOREM

If B is a wf subformula (or sub-wff) of A and A' is the result of replacing zero or more occurrences of B in A by a wff B' and if every free variable of B or B' which is bound in A occurs in the list $y_1, y_2, \ldots, y_k$ then

$$\vdash (y_1)(y_2) \ldots (y_k)(B \equiv B') \supset (A \equiv A') .$$

PROOF: By induction on the number n of connectives and quantifiers in A.

Basis: $n = 0$.

Then A must be an atomic formula, hence either 0 occurrences are replaced or B is A.

If 0 occurrences, then reduces to $C \supset (A \equiv A)$.

If B is A, then reduces to $(y_1) \ldots (y_k)$ $(A \equiv B') \supset (A \equiv B')$. 6

Induction:

We now have $n > 0$ and B a proper part of A.

Assume true for all wffs A with less than n connectives and quantifiers.

Cases: 1. A is $\sim D$.

2. A is $(D \supset E)$.

3. A is $(x)D$.

Case 1. A' is $\sim D'$.

$\vdash (y_1) \ldots (y_k)(B \equiv B') \supset (D \equiv D')$	Ind. hyp.
$\vdash (D \equiv D') \supset (\sim D \equiv \sim D')$	Theorem 2.1
$\therefore \vdash (y_1) \ldots (y_k)(B \equiv B') \supset (\sim D \equiv \sim D')$	Theorem 2.1 and MP

which is the desired result.

Case 2. A' is $D' \supset E'$.

$\vdash (y_1) \ldots (y_k)(B \equiv B') \supset (D \equiv D')$	Ind. hyp.
$\vdash (y_1) \ldots (y_k)(B \equiv B') \supset (E \equiv E')$	Ind. hyp.
$(D \equiv D') \wedge (E \equiv E') \supset ((D \supset E) \equiv (D' \supset E'))$	tautology
$\therefore \vdash (y_1) \ldots (y_k)(B \equiv B') \supset (A \equiv A')$	tantology and MP

Case 3. A' is $(x)D'$.

$\vdash (y_1) \ldots (y_k)(B \equiv B') \supset (D \equiv D')$	
$\vdash (x)[(y_1) \ldots (y_k)(B \equiv B') \supset (D \equiv D')]$	GEN
$\vdash (y_1) \ldots (y_k)(B \equiv B') \supset (x)(D \equiv D')$	Ax 5 and MP by hyp. x not free in $(y_1) \ldots (y_k)(B \equiv B')$
$\vdash (x)(D \equiv D') \supset ((x)D \equiv (x)D')$	13
$\therefore \vdash (y_1) \ldots (y_k)(B \equiv B') \supset ((x)D \equiv (x)D')$	taut. and MP

Corollary Replacement Theorem

A, B, A', B' as above. If $\vdash (B \equiv B')$ then $\vdash (A \equiv A')$.
If $\vdash (B \equiv B')$ and $\vdash A$ then $\vdash A'$.

Corollary Change of Bound Variable

If $(x)B(x)$ is a sub-wff of A and $B(y)$ is similar to $B(x)$ and A' results from A by replacing one or more occurrences of $(x)B(x)$ by $(y)B(y)$,

Then $\vdash A \equiv A'$.

PROOF: By (7) and replacement theorem.

BLANK PAGE

PRENEX NORMAL FORM (PNF)

Note: needed for Completeness Proof to follow.

Note: use as lemmas the useful theorem schemata which are starred.

Definition: A wff C is in prenex (normal form) if C is

$$(Qy_1)(Qy_2) \ldots (Qy_r)\ M$$

where: (1) $y_1, y_2, \ldots, y_r$ are distinct individual vbls., $r \geq 0$, which occur in M,

(2) each (Qy_i) is either (y_i) or $(\exists y_i)$,

and (3) M is a quantifier-free wff.

M is called the matrix of C; $(Qy_1) \ldots (Qy_r)$ the prefix.

Definition: A quantifier (Qy) in a wff is said to be initial if both

(1) (Qy) occurs at the left, or is preceded only by other quantifiers,

(2) the scope of (Qy) extends to the end of the wff.

Corollary: A PNF is a wff in which all quantifiers are non-vacuous and initial.

THEOREM: For any wff C there is a wff C^{o} in PNF such that $\vdash C \equiv C^{o}$.

Note: PNF can be defined so as to be unique.

PROOF: use Church's proof--it gives uniqueness by working on first quantifier not initially placed, provided we use alphabetically earliest possibility when making changes of bound variables.

Procedure:

Let C be written without abbreviations other than existential quantifiers. Starting from left, pick out first non-initial quantifier (Qx) . If there is one, it must be in a wff part of C of one of the forms in column (1)

(1)	(2)		
$\sim(x)B$	$(Ex) \sim B$		14
$\sim(Ex)B$	$(x) \sim B$		15
$A \supset (x)B$	$(x)(A \supset B)$	x not free in A	11
$(x)B \supset A$	$(Ex)(B \supset A)$	x not free in A	16
$A \supset (Ex)B$	$(Ex)(A \supset B)$	x not free in A	17
$(Ex)B \supset A$	$(x)(B \supset A)$	x not free in A	12

The wff's in (2) are equivalent to those in (1), <u>provided</u> x is not free in A . If x is free in A use <u>change of bound variable</u> to a variable not free in A nor occurring in B . Then use replacement theorem.

If quantifiers are not distinct, delete first (Qy_i) by $(Qy_i)(Qy_i)A \equiv (Qy_i)A$ by 9, 10 . Delete (Qy_i) for y_i not in M .

PROOF of termination: By considering the number of connectives not within the scope of the left-most non-initial quantifier and the number of non-initial quantifiers.

PROOF of $\vdash C \equiv C^{o}$. By the lemmas and the replacement theorem, and transitivity of $\equiv$.

Comment: A PNF may contain free variables.

But we can always find closed C' in PNF such that

$$\vdash C \Leftrightarrow \vdash C' .$$

C' is the <u>closure</u> of C.

Remarks on PNF:

Actually need not remove $\wedge$ and $\vee$ but can use

$A \vee (Ex)$	$(Ex)(A \vee B)$	x not free in A
$A \vee (x)B$	$(x)(A \vee B)$	"
$A \wedge (x)B$	$(x)(A \wedge B)$	"
$A \wedge (\exists x)B$	$(\exists x)(A \wedge B)$	"

Examples:

$(x)(F(x) \supset (y)(G(x, y) \supset \sim (z)H(y, z)))$

$(x)(y)(F(x) \supset (G(X, y) \supset \sim (z)H(y, z)))$

$(x)(y)(F(x) \supset (G(x, y) \supset (\exists z) \sim H(y, z)))$

$(x)(y)(F(x) \supset (\exists z)(G(x, y) \supset \sim H(y, z)))$

$(x)(y)(\exists z)(F(x) \supset (G(x, y) \supset \sim H(y, z)))$

$$F_1(x, y) \supset (\exists y([F_2(y) \supset (\exists x)F_2(x) \supset F_3(y))]$$

$$(\exists w)(z)(F_1(x, y) \supset (F_2(w) \supset (F_2(z) \supset F_3(w)))$$

Example with $\vee$, $\wedge$.

A VERY ELEMENTARY SYSTEM W

(Hao Wang)

A Survey of Mathematical Logic, Science Press, Peking, 1963.
(Distributed by North-Holland Publishing Co., Amsterdam.)

The system W contains a single two-place predicate (a dyadic relation) R, three constant names 1, 2, 3 of individuals, and the variables x, y, z, etc. If R holds between x and y, we can write R(x, y). The axioms of W are as follows:

A1. There are exactly the three things 1, 2, 3:

$$(x)(x = 1 \vee x = 2 \vee x = 3)$$
$$\& \; 1 \neq 2 \; \& \; 2 \neq 3 \; \& \; 1 \neq 3$$

A2. R is irreflexive:

$$(x) \sim R(x, x)$$

A3. R is many-one:

$$(x)(y)(z).R(x, y) \; \& \; R(x, z) \supset y = z$$

A4: R is one-many:

$$(x)(y)(z).R(y, x) \; \& \; R(z, x) \supset y = z$$

A5.

$$(x)(Ey)R(x, y)$$

A6.

$$R(1, 2)$$

The concepts of model and satisfiability can be defined thus:

Definition 1. An axiom system is satisfiable if there exists a model or interpretation of the system. An interpretation of an axiom system is an assignment of meanings to the undefined terms of the system according to which all the axioms are true.

In particular, a model of the system W is determined by: (a) a (non-empty) domain D of objects; (b) a rule that associates each constant name with a thing in D; (c) a relation R* as the model of R; (d) a rule of interpretation telling us, for any objects a and b in D, whether R* holds between them, and therefore, derivatively, for any statement, whether it is true or false; (e) the fact that the statements A1-A6 come out true according to (a)-(d).

It is quite easy to find a model for W. Take the domain D as consisting of three persons, Chang, Li, and Yang, sitting around a round table with Chang immediately to the right of Li, associating them with 1, 2, 3, respectively, and interpret the relation R as holding between two persons a and b if and only if a sits immediately to the right of b. It can be checked that all the axioms A1-A6 come out true.

In fact, we can take an arbitrary domain D with three objects 1*, 2*, 3* which represent 1, 2, 3, respectively, and obtain a model for the system W by choosing a relation R* such that R* is true of the pairs (1*, 2*), (2*, 3*), (3*, 1*), and false for the remaining six pairs. As a result, we do not even have to use any concrete interpretations for W. We can say abstractly that the following matrix defines a model for W.

R	1	2	3
1	-	+	-
2	-	-	+
3	+	-	-

We come now to the familiar notion of isomorphism. Thus, two models of W are isomorphic (or essentially the same) if there exists a one-to-one correspondence between the two domains such that the first model of the relation R holds between two objects of the first domain if and only if the other model of the relation R holds between their images in the other domain. It follows that a statement is true in one model if and only if it is true in the other. For instance, any two models for W, which both satisfy the matrix given above, are isomorphic. In general, an axiom system may contain a number of technical terms which stand for properties, relations, and operations. In two isomorphic models of the systems, all these should correspond so that, for example, if f_1 and f_2 stand for a same functor and a_2, b_2 correspond to a_1, b_1, then $f_1(a_1, b_1)$ must correspond to $f_2(a_2, b_2)$ This condition on models for the technical terms is equivalent to the requirement that any statement of the system is true in one model if and only if it is true in the other. We can, therefore, give the definitions:

Definition 2. Two models of an axiom system S are said to be isomorphic if there exists a one-to-one correspondence between the two domains and any statement of S is true in one model if and only if it is true in the other.

Definition 3. An axiom system S is categorical if and only if every pair of models of S is isomorphic.

It is not hard to see that the system W, determined by A1-A6, is categorical. In fact, by straightforward combinatorial considerations, we can see that all models of W satisfy the matrix given above. Thus, by A1, the domain of each model of W consists of exactly three objects (say) 1*, 2*, 3* . Therefore, there are nine ordered pairs of the objects of that domain. For each of these pairs, R may either hold or not. Hence, we have 2^9 possible interpretations

of the relation R which would all satisfy A1. By A2, R(1*, 1*), R(2*, 2*), R(3*, 3*), must all be false. Therefore, there are only $2^6 (=2^9/2^3)$ possible interpretations of R satisfying both A1 and A2. Of these 64 possibilities, only 27 satisfy also A3, because, by A3, if R holds of the pair (1, 2) then it cannot hold of (1, 3), and so on. By similar considerations, we see easily that of the 27 remaining possibilities, only 18 satisfy also A4, 2 satisfy A4 and A5, and only one satisfies A4-A6. This interpretation of R that satisfies all the axioms A1-A6 is determined by the matrix already given: R is true for the pairs (1*, 2*), (2*, 3*), (3*, 1*), and false for the six remaining pairs. Hence, W is categorical.

Thus, it is clear that additional axioms serve, in general, to reduce the number of permissible distinct interpretations for a system. When we add enough axioms to reduce the number of interpretations to one (up to isomorphism), we have a categorical system. But if we add any more axioms which would eliminate also the last interpretation, the resulting system would not be satisfiable according to Df. 1.

In fact, once we assume A1, the problem of finding additional axioms to obtain a categorical and satisfiable system is pretty trivial. For example, instead of A2-A6, we can use directly the following:

A1*. R is true of the pairs (1, 2), (2, 3), (3, 1) and false for the six remaining pairs consisting of 1, 2, and 3.

A1 and A1* determine the same interpretation as A1-A6. Or, we can choose any one of the other possible interpretations of R by using some analogous axiom in place of A1*. Then we would have in each case a different system, which is again categorical.

If we omit from W the names 1, 2, 3, then we can no longer state the axioms A1 and A6, although we can still keep the axioms A2-A5. In place of A1, we can use:

A1'. There exist only three distinct things: $(Ex)(Ey)(Ez)(w)(x \neq y \,\&\, y \neq z \,\&\, x \neq z \,\&\, (w = x \vee w = y \vee w = z))$.

But nothing resembling A6 can be expressed in the new system. The system determined by A1' and A2-A5 can again be shown to be categorical and complete; the lack of anything like A6 is compensated by the decrease in expressing power caused by the omission of the names 1, 2, 3.

Furthermore, if we use instead of the relation symbol R a function symbol f, then we can replace A2-A5 by the following:

A2'. $(x)(f(x) \neq x)$.

A3'. $f(y) = f(z) \supset y = z$.

The system determined by A1'-A3' is essentially the same as the system determined by A1' and A2-A5; in the new formulation, A4 and A5 become absorbed into elementary logic and notational conventions.

Since W has a model, W is satisfiable.

Definition 4. A system is said to be complete if every proposition (closed wff) p in the system is either provable or refutable; in other words, for every p, either p or $\sim p$ is a theorem.

From Df. 3 and Df. 4, we can prove:

Theorem 1. Every categorical system is complete.

If a system is not complete, there is a proposition p in the system such that neither p nor $\sim p$ is a theorem. Hence by Th. 10 given below in §7,* there exists one model in which p is true and one model in which p is false. These two models cannot be isomorphic. Hence, the system cannot be categorical.

---

*Theorem 10. A system formulated in the predicate calculus without equality is consistent iff it is satisfiable. (Skolem, Herbrand, Gödel.)

Since W is categorical, it is complete.

One may also regard the choice of a model as the construction of a sort of truth definition for the system under consideration, specifying the propositions which are true under the interpretation. In fact, in every case we require that all theorems must be true in the model and that for every proposition p, either p or $\sim p$ but not both must be true. Hence, when a system is complete, the theorems must coincide with the true propositions. It follows that for a complete system, a decision procedure for provability also yields a decision procedure for truth, and (conversely).

Definition 5. A decision procedure for provability (truth, validity) of an axiomatic system is an effective method by which, given any proposition of the systems, we can decide in a finite number of steps whether it is provable (true, valid).

In the case of the system W which has only one model, we can easily give at once a decision procedure for both truth and provability. Thus, after eliminating " $\wedge$ ", " $\supset$ ", "(Ex)" in familiar manner, we can characterize all propositions of the system W:

(i) If a and b are numbers among 1, 2, and 3, then Rab and $a = b$ are (atomic) propositions;

(ii) If p and q are propositions, so are $\sim p$ and $(p \vee q)$.

(iii) If Ha is a proposition, so is (x)Hx .

(iv) There are no other propositions except those required by (i) - (iii).

A truth definition is simply:

(i) Among the atomic propositions, R12, R23, R31, $1 = 1$, $2 = 2$, $3 = 3$ are true, all others are false;

(ii) $\sim p$ is true if and only if p is false, $(p \vee q)$ is true if and only if either p or q is true;

(iii) $(x)Hx$ is true if and only if H1, H2, H3 are true.

This truth definition gives a decision method because for every proposition of W, no matter how complex, we can always reduce the question of its truth to that of less complex propositions, in such a way that in a finite number of steps we arrive at a finite number of atomic propositions which can be decided by (i).

Hence, there is a decision procedure for W both for truth and for provability.

If we delete A6 from W, the resulting system is no longer complete, but we can easily see that it still has a decision method for provability.

<u>Theorem 2</u>. There exist incomplete axiom systems for which there are decision procedures for provability.

Incidentally, the axioms A1 and A1' have a different character from the other axioms in so far as they do not assert properties of R and f but directly specify their domain. Such axioms are sometimes called "axioms of limitation".

<u>Definition</u>. Given an axiomatic theory, a subset X of the axioms is said to be <u>independent</u> if some wff in X cannot be proved from the rest of the axioms.

SATISFIABILITY

Relations

Binary relations [illegible]

a binary relation [illegible] each element [illegible]

For present purposes binary relations [illegible]

relations are needed, for [illegible]

If X is a set and k [illegible] denote by X^k the set of all [illegible] of elements $b_1, \ldots, b_k$ [illegible] X [illegible] the Cartesian product of X with [illegible]

An n-place (or n-ary) relation [illegible] set of X^n, i.e., a set of ordered n-tuples [illegible] of X. For example, the 3-place relation [illegible] between points on a line is the set of all 3-tuples (x, y, z) such that the point x lies between the points y and z. A [illegible] relation is called a binary relation, i.e., the binary relation of fatherhood is the set of all ordered pairs (x, y) such that x and y are human beings and x is the father of y. A 1-place relation on X is a subset of X, and is called a property on X.

Interpretation

An interpretation of a wff A is a non-empty set D (the domain of the interpretation) and an assignment to each n-adic predicate letter of A of an n-ary relation in D

Example

A1: $(x)(F(x) \supset G(x)) \supset ((x)F(x) \supset (x)G(x))$

Take domain as the positive integers.

Take F as the property 'x is divisible by 4'

i.e., the set of multiples of 4 .

Take G as the property 'x is divisible by 2'

i.e., the set of all even numbers.

Then, under the interpretation, A_1 is <u>true</u>.

Example

A_2: $(Ex)F(x) \supset (Ey)G(y)$

Then under the same interpretation A_2 is true.

But under the interpretation which follows, it is false.

Take D as $\{0, 1\}$.

F as the property $x = x$ i.e.,

the set $D = \{0, 1\}$.

G as the property $x \neq x$ i.e.,

the empty set $= \{\} = \emptyset$.

Example

A_3: $(y)F(x, y)$

Take same domain.

Take $F(x, y)$ as $x \leq y$.

Then <u>under the interpretation</u> A_3 is neither true nor false--it represents the 1-ary relation

$$(y)(x \leq y)$$

and is true for $x = 1$, false otherwise. Note that A_3 is not closed.

A wff A is <u>satisfiable</u> if there is some interpretation of A (with non-empty domain D) <u>and</u> some assignment of elements of D to the free variables of A which make A true.

<u>Examples</u>

A_1 is satisfiable.

A_2 is satisfiable.

A_3 is satisfiable (assign 1 to x).

A wff A is <u>valid</u> if under <u>every</u> interpretation and <u>every</u> assignment of elements of the domain D of the interpretation to its free variables A is true. (This is eqv. to previous def.).

<u>Examples</u>

A_1 is valid.

A_2 is not valid.

A_3 is not valid.

A closed wff is either <u>true</u> or <u>false</u> under any given interpretation.

COROLLARY A wff A is valid iff $\sim A$ is not satisfiable.

The notion <u>valid</u> corresponds to <u>always true</u>. (No counter-example).

DEFINITION: The closure of a wff A with free variables

$$y_1, \ldots, y_m \text{ is } (y_1) \ldots (y_m)A .$$

[Remark: By ② the order of these universal quantifiers doesn't matter.]

COROLLARY A is valid iff the closure of A is valid.

THEOREM (Soundness)

Every theorem of the first-order predicate calculus is valid.

PROOF: Axioms are valid.

Rules of inference preserve validity.

(See details on pages 79-81.)

COROLLARY To show A is not a theorem, it suffices to show A not valid, i.e., $\sim A$ satisfiable.

Validity and Theoremhood

Given a wff A, suppose we are concerned with whether or not A is a theorem. If we can prove A, then A is a

theorem (and is valid). But if we do not succeed in proving A, perhaps A is not a theorem. To show that A is not a theorem, (since it is possible that neither A nor $\sim A$ is a theorem because pred. calc. not complete in that sense) it suffices to show that A is not valid, that is $\sim A$ is satisfiable. By the Gödel Completeness Theorem (to be proved) A is a theorem <u>iff</u> A is valid. Hence it will be the case that either A is a theorem or $\sim A$ is satisfiable. And while this is not enough to yield a decision procedure, it will be enough to yield an effective proof procedure. (This is an alternative proof procedure to the purely syntactic one already given for all formal theories.)

Proof procedures and decision procedures

THEOREM:

In any axiomatic formal theory, there is an effective proof procedure.

PROOF:

1. Reduce countable (finite or enumerable) set of symbols to finite set by use of subscript 1 as new symbol. I.e., replace A_1, A_2, ..., by A_1, A_{11}, ... (prove unique readability.)
2. Introduce a new symbol called carriage return.
3. Now we could enumerate all expressions composed of the finite set of symbols as follows:

   All expressions consisting of one occurrence of a symbol.

   All expressions consisting of two occurrences of symbols.

   ...

   All expressions consisting of n symbol occurrences.

   ...
4. Now, since wffs are effective, we could redo the enumeration...saving in the list only expressions which are wffs (between carriage return symbols).
5. Now, since proofs are effective (axioms are, and rules of inference are) we can redo the list so that every string in the list is a proof.

6. Eventually every proof will occur in our list.

7. A proof is a proof <u>of</u> its last line.

8. Therefore to find a proof for a given well-formed formula A we need only construct the list until at last we come to a proof of A .

9. <u>But</u>, if A is not a theorem, we will never know that we should give up, hence we will go on forever.

Details of Soundness Proof

Axioms are valid

Axiom 1. $A \supset (B \supset A)$

To satisfy the negation we must find an interpretation and an assignment to the free vbls. which makes A true, B true and A false. Clearly impossible.

Axiom 2. $(A \supset (B \supset C)) \supset ((A \supset B) \supset (A \supset C))$

To satisfy the negation we must have

$A \supset (B \supset C)$	true
$A \supset B$	true
$A \supset C$	false

hence

A	true
C	false
B	false
A	false

Axiom 3. $(\sim B \supset \sim A) \supset ((\sim B \supset A) \supset B)$

To satisfy the negation must have

	$\sim B \supset \sim A$	true
	$\sim B \supset A$	true
	B	false
hence	$\sim B$	true
hence	$\sim A$	true
	A	true

Axiom 4. $(x)A(x) \supset A(y)$ $\quad$ y free for x in $A(x)$

To satisfy the negation must have

$(x)A(x)$	true
$A(y)$	false

So suppose $d \in D$ assigned to y and $\sim A(d)$. Then $\sim(x)A(x)$. Note role of proviso on y . Otherwise there is no free variable in $A(y)$.

Axiom 5. $(x)(A \supset B(x)) \supset (A \supset (x)B(x))$ $\quad$ x not free in A

To satisfy negation must have

$(x)(A \supset B(x))$	true	
A	true	
$(x)B(x)$	false	$\therefore$ for some d, $\sim B(d)$

Hence have $\sim(A \supset B(d))$ which contradicts (1) .

Rules of inference preserve validity

Modus Ponens

$$\frac{\begin{array}{l} A \supset B \\ A \end{array}}{B}$$

Suppose $A \supset B$, A are valid and B is not valid. $\sim B$ is satisfied by an interpretation $\mathcal{I}$ with assignment $(d_1, \ldots, d_n)$ to the free variables $(x_1, \ldots, x_n)$. Then this assignment makes $A \supset B$ true (since $A \supset B$ is valid) hence makes A false, hence contradicts validity of A.

Generalization

$$\frac{A}{(x)A}$$

If $(x)A$ is not valid then for some interpretation $\mathcal{I}$ and some $d \in D$, $\sim A(d)$. Hence A not valid.

Notice that there are now 3 kinds of equivalences:

1	$\vdash A \equiv B$	equivalence
2	$\vdash A \Leftrightarrow \vdash B$	interprovability
3	A valid $\Leftrightarrow$ B valid	intervalidity

1 $\Rightarrow$ 2 REPLACEMENT THEOREM

2 $\Rightarrow$ 3 SOUNDNESS & COMPLETENESS (to be proved)

<u>Examples</u>

1. $\vdash (x)F(x) \Leftrightarrow \vdash (Ey)G(y)$

since both sides are <u>false</u>

but certainly <u>not</u>:

$$\vdash (x)F(x) \equiv (Ey)G(y)$$

for this is <u>not</u> <u>valid</u>. Take domain $\{0, 1\}$,

take $G(y)$ as $y \neq y$

take $F(x)$ as $x = x$.

2. $\vdash (x)F(x) \Leftrightarrow \vdash (x) \sim F(x)$

<u>not</u> $\vdash (x)F(x) \equiv (x) \sim F(x)$

Example using the closure of a *wff* .

<u>Interprovability</u> $\vdash F(x) \Leftrightarrow \vdash (x)F(x)$

PROOF:

<u>L to R</u>: Suppose $\vdash F(x)$. Then we construct a proof of $(x)F(x)$ as follows:

$$\left.\begin{array}{c} \vdots \\ F(x) \end{array}\right\} \quad \text{proof of } F(x)$$

$$(x)F(x) \qquad \text{GEN}$$

<u>R to L</u>: Suppose $\vdash (x)F(x)$. Then we construct a proof of $F(x)$ as follows:

$$\left.\begin{array}{c} \vdots \\ (x)F(x) \end{array}\right\} \quad \text{proof of } (x)F(x)$$

$$(x)F(x) \supset F(x) \qquad \text{Axiom 4}$$

$$F(x) \qquad \text{MP}$$

<u>Intervalidity</u> $F(x)$ valid $\equiv$ $(x)F(x)$ valid

By Corollary above (from definition of <u>valid</u>).

<u>Equivalence</u>

<u>not</u> $\vdash F(x) \equiv (x)F(x)$

<u>Note</u>: We have proved this if x is not free in $F(x)$.

PROOF:

Every theorem is valid (soundness). Therefore it suffices to show that

$$F(x) \equiv (x)F(x)$$

is not valid. We construct an interpretation under which it is not true.

Take $D\ \{0, 1, 2\}$.

Take $F(x)$ as a relation (property) which is true for 1, false for 0, 2:

F(0)	F(1)	F(2)
false	true	false

(Example might be oddness.)

Take 2 as the assignment to the free vbl. x .

Then under this interpretation the formula is <u>true</u> iff

$$F(2) \equiv F(0) \wedge F(1) \wedge F(2)$$

so true under this interpretation.

But now take 1 as free vbl. Then under the new interpretation

$$F(1) \equiv F(0) \wedge F(1) \wedge F(2)$$

which is <u>false</u>. (Satisfies $F(x) \not\equiv (x)F(x)$.) <u>Hence not valid</u>.

SKOLEM NORMAL FORM

DEFINITION. A wff A is in Skolem normal form (SNF) if it is in closed prenex normal form with prefix

$$(Ey_1) \ldots (Ey_m)(z_1) \ldots (z_n) \cdot \qquad m, n \geq 0$$

TO BE PROVED.

For every wff A there exists a wff A' in SNF such that

$$\vdash A \Leftrightarrow \vdash A'$$

and

$$A \text{ valid} \Leftrightarrow A' \text{ valid.}$$

Temporary notation:

$\overrightarrow{(Ey)}$ for $(Ey_1) \ldots (Ey_n)$

$\vec{y}$ for $y_1, \ldots, y_n$

$A[\bar{y}, u]$ to exhibit <u>all</u> the free variables of the wff A.

Let A be $\overrightarrow{(Ey)}(u)B[\vec{y}, u]$

A_1 be $\overrightarrow{(Ey)}((u)(B[\vec{y}, u] \supset F(\vec{y}, u)) \supset (u)F(\vec{y}, u))$

where B has $\vec{y}$, u as its only free vbls, and F is an $n+1$-adic predicate letter not in A.

Show how this will lead to SNF.

LEMMA 1 $\qquad \vdash A_1 \Rightarrow \vdash A$

PROOF:

$$\vdash (\overrightarrow{Ey})\left(\left(u\right)\left(B[\vec{y}, u] \supset F(\vec{y}, u)\right) \supset (u)F(\vec{y}, u)\right)$$

Assume given a proof of A_1; from it we construct a proof of A.

Take $B^*[\vec{y}, u]$ as the result of replacing all bound vbls of $B[\vec{y}, u]$ by new vbls. which do not occur in the given proof. Replace $F(\vec{z}, w)$ <u>throughout</u> the proof by $B^*[\vec{z}, w]$. Show the result is a proof.

Instances of Axioms 1, 2, 3, MP and GEN all okay. Instances of Ax. 4 $[(x)A(x) \supset A(w)$ provided w free for x in $A]$ all okay since all new quantifiers have new vbls hence do not have w. Instances of Ax. 5 $[\ (x)(A \supset B(x)) \supset (A \supset (x)B(x))$ provided x not free in $A\ ]$ all okay since $B^*[\vec{z}, w]$ has same free vbls as $F(\vec{z}, w)$. Hence, by construction

$$\vdash (\overrightarrow{Ey})((u)(B[\vec{y}, u] \supset B^*[\vec{y}, u]) \supset (u)B^*[\vec{y}, u]) .$$

Now, by change of bound variables, the asterisks can be removed to give:

$$\vdash (\overrightarrow{Ey})((u)(B[\vec{y}, u] \supset B[\vec{y}, u]) \supset (u)B[\vec{y}, u])$$

Let G, w be new:

$B \supset B \equiv G(w) \vee \sim G(w)$	Theorem 2.1 (prop. calc.)
$(\overrightarrow{Ey})[(u)(G(w) \vee \sim G(w)) \supset (u)B[\vec{y}, u]]$	replacement theorem
$(\overrightarrow{Ey})[(G(w) \vee \sim G(w)) \supset (u)B[\vec{y}, u]]$	Theorem 9 and replacement
$G(w) \vee \sim G(w) \supset (\overrightarrow{Ey})(u)B[\vec{y}, u]$	Theorem 17 and replacement
$G(w) \vee \sim G(w)$	Theorem 2.1
$(\overrightarrow{Ey})(u)B[\vec{y}, u]$	MP

LEMMA 2 $\qquad \vdash A \Rightarrow \vdash A_1$

Insert to proof of Skolem normal form (Mendelson p. 89). The following replaces the argument from line 2 "Conversely, ..." to line 8 "... $\vdash A_1$ ". The replacement is needed to avoid use of Rule C and individual constants. To prove:

$$\vdash A \Rightarrow \vdash A_1$$

where A is

$$(Ey_1) \ldots (Ey_n)(u)B[y_1, \ldots, y_n, u]$$

and A_1 is

$$(Ey_1) \ldots (Ey_m)([(u)(B \supset F^{n+1}(y_1, \ldots, y_n, u))]$$

$$\supset (u)F^{n+1}(y_1, \ldots, y_n, u))$$

PROOF:

1. For any wffs C, D

   $(x)C(x) \supset ((x)(C(x) \supset D(x) \supset (x)D(x)$ By theorem 3 and prop. calc.

2. $(u)B \supset ((u)(B \supset F^{n+1}) \supset (u)F^{n+1})$ Instance of 1.

3. For any wffs C, D

   $(x)(C(x) \supset D(x)) \supset ((Ex)C(x) \supset (Ex)D(x))$ Thm. 4

4. $(y_n)[(u)B \supset ((u)(B \supset F^{n+1}) \supset (u)F^{n+1})]$ GEN of 2.

5. $(Ey_n)(u)B \supset (Ey_n((u)(B \supset F^{n+1}) \supset (u)F^{n+1})$

   4, instance of 3, MP.

Repeating steps 4, 5 with $y_{n-1}, \ldots, y_1$, we obtain

6. $(Ey_1) \ldots (Ey_n)(u)B \supset (Ey_1) \ldots (Ey_n)((u)(B \supset F^{n+1})$

   $\supset (u)F^{n+1})$

Hence, by the hypothesis $\vdash A$ and MP

$$\vdash (Ey_1) \ldots (Ey_n)((u)(B \supset F^{n+1}) \supset (u)F^{n+1})$$

that is, $\vdash A_1$.

THEOREM For every wff A we can effectively find a wff A' in Skolem normal form such that $\vdash A$ iff $\vdash A'$.

PROOF

1. By previous theorem we can find A^{o} in closed prenex normal form such that $\vdash A \equiv \vdash A^{o}$.
2. Now by the construction given above we can, at each step, reduce by one the number of existential quantifiers which precede universal quantifiers. From

$$(\overrightarrow{Ey})(u)\ B(\vec{y},\ u)$$

where

$$B(\vec{y},\ u) \text{ is } (\overrightarrow{Qz})\ B''[\vec{y},\ \vec{z},\ u]$$

we get

$$(\overrightarrow{Ey})((u)(B(\vec{y},\ u) \supset F(\vec{y},\ u)) \supset (u)F(\vec{y},\ u))$$

which gives

$$(\overrightarrow{Ey})(Eu)(B'(\vec{y},\ u,\ w)) \quad \text{which is}$$

$$(\overrightarrow{Ey})(Eu)(\overrightarrow{Qz})\left[(w)B''(\vec{y},\ u,\ \vec{z},\ w)\right]$$

where w is new and where the quantifier (w) is right-most in the prefix of B'.

NOTE THE TRADE-OFF

For each universal we get one new existential.

Hence, for example, $\exists\forall\exists$ becomes $\exists\exists\exists\forall$. This is important in the consideration of reduction classes.

In addition, since we plan to use the Skolem normal form to prove completeness, we need

$$A \text{ valid} \Leftrightarrow A' \text{ valid} .$$

(Mendelson has already proved completeness, hence does not need this step.)

THEOREM: A wff A is valid in a given non-empty domain iff its Skolem normal form is valid in that domain. A wff is valid iff its Skolem normal form is valid.

PROOF: Parallels the proof that $\vdash A \Leftrightarrow \vdash A'$, except that wherever that proof makes use of a theorem, the present proof makes use of the fact that the theorem is valid, and wherever that proof makes use of a rule of inference the present proof must instead use the fact that the rule of inference preserves validity (in an arbitrary non-empty domain).

<u>Example 1 for Skolem normal form</u>

$$(x)(G(x) \supset H(x)) \supset ((\exists x)G(x) \supset (\exists x)H(x))$$

Put into PNF. Already closed. Working L to R we would get

(1) $(\exists x)(y)(\exists z) \,.\, (G(x) \supset H(x)) \supset (G(y) \supset H(z))$

But if we chose to pull out the quantifiers in a different order, we could get SNF immediately:

(2) $(\exists x)(\exists y)(z) \,.\, (G(x) \supset H(x)) \supset (G(z) \supset H(y)) \,.$

To put (1) in SNF

$$(\exists x) \,.\, (y)((\exists z)((G(x) \supset H(x)) \supset (G(y) \supset H(z))) \supset F(x, y))) \supset (y)F(x, y)$$

$$(\exists x)(\exists y)(\exists z)(u)(((G(x) \supset H(x)) \supset (G(y) \supset H(z))) \supset F(x, y)) \supset F(x, u)$$

i.e.

$$(\exists x)(\exists y)(\exists z)(u)(G(x) \supset Hx \supset (G(y) \supset H(z)) \supset F(x, y) \supset F(x, u))$$

N.B. We can<u>not</u> always obtain SNF directly.

Proof: $\exists\forall\exists$ is unsolvable--All other prefixes with 3 quantifiers are solvable.

<u>Example 2 for Skolem normal form</u>

Mendelson, page 89.

$$(x)(y)(\exists z)\ A(x, y, z)$$

where A is a quantifier-free wff with x, y, z as its only free variables.

Note: in this case we start with a universal quantifier.

Note: predict that final answer will have prefix $\exists\exists\exists\forall\forall$.

$$(x)((y)(\exists z)\ A(x, y, z) \supset F(x)) \supset (x)F(x)$$

where F is new. Now put in PNF:

$$(\exists x)(y)(\exists z)(v)\ .\ (A(x, y, z) \supset F(x)) \supset F(v)$$

Let this be

$$(\exists x)(y)(\exists z)(v) \qquad B(x, y, z, v)\ .$$

Note that B is a quantifier-free wff. $(\exists z)(v)B$ has x and y free.

$$(\exists x)((y)((\exists z)(v)\ B(x, y, z, v) \supset G(x, y)) \supset (y)G(x, y))$$

where G is new .

$$(\exists x)(\exists y)(\exists z)(v)(w)\ .\ (B(x, y, z, v) \supset G(x, y)) \supset G(x, w)$$

which is

$$(\exists x)(\exists y)(\exists z)(v)(w)\ .\ ((((A(x, y, z) \supset F(x)) \supset F(v)) \supset G(x, y) \supset G(x, w))$$

Note prefix.

Example 3 for Skolem normal form

$$(\exists x)(\exists y)\ F(x,\ y,\ z) \supset (\exists y)(\exists z)\ F(y,\ z,\ x)$$

Put into PNF.

$$(\exists y)(\exists u)(v)(w)\ .\ F(v,\ w,\ z) \supset F(y,\ u,\ x)$$

Now need to get closed PNF.

$$(z)(x)(\exists y)(\exists u)(v)(w\ .\ F(v,\ w,\ z) \supset F(y,\ u,\ x)$$

$$(z)((x)(\exists y)(\exists u)(v)(w)A \supset G(z)) \supset (z)G(z)$$

$$(\exists z)(x)(\exists y)(\exists u)(v)(w)(x_1)((A \supset G(z)) \supset G(x_1))$$

$$(\exists z)(\exists x)(\exists y)(\exists u)(v)(w)(x_1)(x_2)$$

$$((((A \supset G(z)) \supset G(x_1)) \supset H(z,\ x)) \supset H(z,\ x_2))$$

Or, using the left parenthesis convention

$$\ldots \quad A \supset G(z) \supset G(x_1) \supset H(z,\ x) \supset H(z,\ x_2)\ .$$

BLANK PAGE

THE INFINITY LEMMA

There are a group of results which are closely connected with the famous infinity lemma, which can be stated thus:

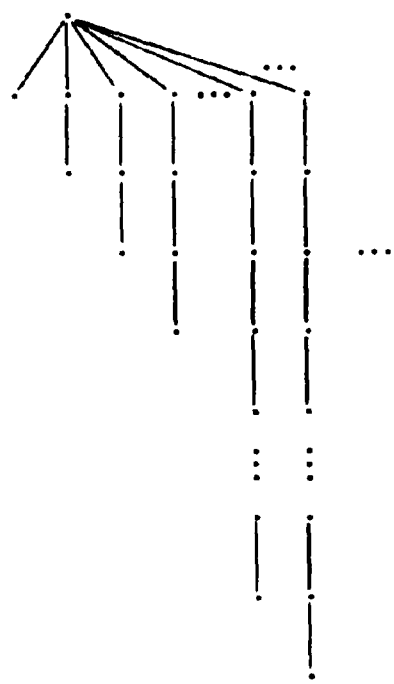
<u>The Infinity Lemma</u>

> If there is an infinite sequence $Q_1, Q_2, \ldots$ of disjoint finite sets of ordered pairs of points such that the first point of each pair in Q_{i+1} $(i = 1, 2, \ldots)$ is the same as the second point of some pair in Q_i, then there is an infinite sequence of points $P_1, P_2, \ldots$ such that (P_i, P_{i+1}) belongs to Q_i, for every i.

Consider the set of finite paths each of which consists of a member of Q_1 followed by a member of Q_2, and so on. The set is infinite since each member of each Q_i, for every i, occurs as the last edge of some finite path. Hence, there must be at least one pair (P_1, P_2) in Q_1 which occurs in infinitely many finite paths. All these finite paths must contain as the second edges finitely many (P_2, P_3) in Q_2, and hence there must be some P_3 such that there are infinitely many finite paths which begin with (P_1, P_2) and are followed by (P_2, P_3). Continuing thus, we get the desired infinite sequence $P_1, P_2, \ldots,$ which makes up an infinite path.

To emphasize the nontrivial character of the infinity lemma, consider a case where one point of level one is

connected to an infinite number of points $A_1, A_2, A_3, \ldots$ on level two such that A_i goes to the $(i+1)$th level. In such an example, there exists no infinite path.



<u>Lemma</u> <u>Law of Infinite Conjunction</u>

Let $A_1, A_2, \ldots$ be an infinite sequence of propositional formulas such that for every n, there is an assignment of truth-values which makes $A_1, A_2, \ldots, A_n$ simultaneously true. Then there is an assignment which makes all of $A_1, A_2, \ldots$ simultaneously true.

PROOF: Using infinity lemma.

König's Infinity Lemma

D. König, Theorie der endlichen and unendlichen Graphen, Leipzig, 1936, (Reprinted by Chelsea, 1950), pages 81-85.

Translated by Anthony Sholl

Thm 6: (Unendlichkeitslemma): Let $\pi_1, \pi_2, \pi_3, \ldots$ be a denumerably infinite sequence of finite, nonempty, pairwise-disjoint point sets. The points contained in the union of these sets are taken as the nodal points of a graph G. If now G has the property that each point of $\pi_{n+1} (n = 1,2,3\ldots)$ is connected to some point of π_n by an edge* of G, then G has at least one simple**, infinite path $P_1 P_2 P_3 \ldots$, where P_n $(n = 1,2,3\ldots)$ is a point in π_n.

For the proof of this theorem we shall call a (finite) path in G an S-path if its $\{^{\text{points}}_{\text{nodes}}\}$ belong by turns to $\pi_1, \pi_2, \ldots, \pi_k$. There are infinitely many S-paths in G, in fact, with the exception of the points of π_1, every node of G is the terminus of some S-path. Each S-path begins with an edge which connects a given point P_1 in π_1 with a point X_2 in π_2. Since there exist only a finite number of such edges, one of the edges, say $P_1 P_2$, must occur in infinitely many S-paths. All of these S-paths contain as their second edges one of the finitely many edges P_2X_3 where X_3 belongs to π_3; hence, there must be in π_3 a point P_3 with the property that infinitely many S-paths which begin with P_1P_2 also contain P_2P_3. Continuing similarly, one defines a point P_4 in π_4, P_5 in π_5, and so on. The process cannot terminate, and it leads to an infinite path $P_1P_2P_3\ldots$ of the desired type.

---

* By an edge of G is meant any path of length one between two nodes of G.

**A path $P_1P_2P_3\ldots$ is simple if for $i \neq j$, $P_i \neq P_j$.

The infinity lemma proved here[1] lends itself to applications not only in graph theory - of which we shall give many examples later on - but also in the various mathematical desciplines where often it provides a useful method for extending certain results from the finite to the infinite domain. Three examples follow.

The first example concerns kindredship, which, in the form of the genealogical tree, provides an old and well known application of graphs. We show namely that if one takes as hypothesis that mankind will never become extinct, then there exists some person, alive now, who is the ancestor of an unending line of descendents[2].

Let E_1 be the set of persons alive at this moment; E_2 the set of children of members of E_1; E_3 the set of children of members of E_2; and so on. By the hypothesis above - and because of the finiteness of human life - none of the sets $E_1, E_2, E_3, \ldots$ is empty. Since a given person can have only finitely many children, it follows from the finiteness of E_1 that all the sets E_i are finite. With each element in a given set E_i let us associate a point so that the set E_i and the point set π_i correspond one-to-one $(i = 1,2,3,\ldots)$.[3] We take the points of these sets π_i as the nodes of a graph G. A node A from π_{n+1} will be connected by an edge of G to a node B from π_n if the person corresponding to the point A is child of the person corresponding to point B. Other edges are not admitted. The graph G so defined and the sets π_i satisfy the conditions of the Infinity Lemma. Application of the lemma yields, therefore, an endless sequence $a_1, a_2, a_3, \ldots$ with the property that a_i is an element of E_i and a_{i+1} is a child of a_i. Consequently, a_1 is a contemporary person of the desired type.

---

(1) This proof, as does the one above of theorem 3, uses the axiom of choice. In most applications of the infinity lemma, however, the use of the axiom can be avoided. We shall not go into the matter further here.
(2) This says more than the assertion simply that there exists a person, alive now, who has infinitely many descendents. That goes without saying.
(3) An individual can belong to more than one of the sets E_i. In that case we let him correspond to different points according as he is construed as a member of one or another of the "generations" E_i; the π_i are then disjoint.

By a similar consideration, one can show that the existence of an endless male line follows from the interminability of the male sex.

Many instances of application of the Infinity Lemma are applications of an analogue of the Borel covering theorem. It seems interesting to notice, therefore, that from one point of view the Infinity Lemma may be conceived as the real basis of these "Borelish" theorems. We shall proceed to reduce to the Infinity Lemma the following theorem of de la Vallee Poussin which clearly subsumes Borel's theorem as a special case:

> Let E be a closed subset of the interval $(0, 1)$ and I, a set of intervals so constituted that each point of E is contained in some one of these intervals. Then there is a natural number n such that if one partitions $(0,1)$ into 2^n equal subintervals, those subintervals which contain a point of E are (themselves) included in some interval belonging to the set I.

If the theorem were false, then for each $<$ value of $> n$, there would be at least one subinterval $(\frac{m}{2^n}, \frac{m+1}{2^n})$, where m is 0 or 1, or 2, or $\ldots$, or $2^n - 1$, which contains a point of E and is included in no interval belonging to I. We designate the set of these subintervals by E_n. With each element of the set E_i we associate a point in such a manner that E_i and the point set π_i are in one-to-one correspondence $(i = 1,2,3,\ldots)$. We take the points of these sets π_i as the nodes of a graph G. A node A from π_{n+1} is connected by an edge of G to a node B from π_n in case the interval corresponding to A arises from the interval corresponding to B by bisection; other edges are not admitted. The graph G so defined and the point sets π_i satisfy the conditions of the Infinity Lemma. Application of the lemma gives the following result. There exists an endless sequence of intervals, $a_1, a_2, a_3, \ldots$ which all

1° Arise from predecessors by bisection;

2° Contain a point of E

3° Are included in no interval contained in I.

Then, however, the point α common to the intervals $a_1, a_2, a_3, \ldots$ is contained in no interval which is a member of I. But that is impossible because by the closure of E, α belongs to E. (This proof makes use only of the theorem on nested intervals, not of the Bolzano-Weierstrass theorem, and it remains valid for the plane, 3-space, etc.).

The third application of the Infinity Lemma is based on the following so-called Baudet conjecture proved by van der Waerden.

α) If k and l are two arbitrary natural numbers, then there is a number N (which depends on k and l) with the property that however one partitions the set $1,2,\ldots N$ into k pairwise disjoint parts, one of these parts contains an l-termed arithmetic progression.

We do not prove this theorem here but show that it is equivalent to the following theorem:

β) If k and l are arbitrary natural numbers and if one partitions the totality of natural numbers entirely arbitrarily into k pairwise disjoint parts, at least one of these parts contains an l-termed arithmetic progression.

It is clear that β) follows from α). The converse of this assertion goes through with the help of the Infinity Lemma as follows. We consider as the set E_n those partitions of the set $Z_n = 1,2,\ldots, n$ into k disjoint parts which are so constituted that none of the < corresponding > k parts contains an l-termed arithmetic progression; E_n is, of course, finite. If we assume that theorem α) is false, then none of the sets E_n is empty. We associate points with the elements of the sets E_n in such a way that the sets E_n and the point sets π_n are in one-to-one correspondence $(n = 1,2..)$. A point of π_{n+1} is connected by an edge to a point π_n if the corresponding elements A of E_{n+1} and B of E_n stand in the following relation. The partition B of Z_n arises from the partition A of Z_{n+1} by the deletion of the number "n+1". The graph so defined and the sets π_i satisfy the conditions of the Infinity Lemma, which applied, yields an endless sequence $A_1, A_2, A_3,$ with the property that, for each

1° A_n is an element of E_n;

2° Two numbers which belong to the same block of the partition A_n also belong to the same block of the partition A_{n-1} (therefore also to the same blocks of $A_{n-2}, A_{n-3}, \ldots$) .

If one assigns each pair of natural numbers to the same class if and only if these two numbers belong to the same block of some partition A_n (therefore to the same block of all partitions A_n in which the two numbers appear) he obtains a partition of the natural numbers into k disjoint parts < where the blocks of this partition are the "classes" cited above > . By theorem β) one of these blocks contains an ℓ-termed arithmetic progression. If N is the largest number of this progression then this sequence must already be contained in <u>some</u> block of the partition A_N which belongs to the set E_N . This condition contradicts the definition of the sets E_n . (One sees that this proof of the equivalence of theorems α) and β) remains valid when instead of arithmetic progressions other classes of finite sets of numbers are taken into consideration, for example for geometric progressions, etc.)

BLANK PAGE

Gödel Completeness Theorem (See also Church §44)

THEOREM: Every valid wff of the first-order predicate calculus is a theorem

PROOF:

A wff is valid iff its Skolem Normal Form is valid, provable iff its Skolem Normal Form is provable. Therefore, it suffices to consider only formulas in Skolem normal form. Further, we may assume that the first quantifier is an existential, since if not, (Ey), where y is new, can be prefixed.

OUTLINE OF PROOF:

From A we will construct a sequence of formulas B_1', B_2', ... of the propositional calculus such that:

(a) If for some k, $B_1' \vee \ldots \vee B_k'$ is a tautology, A is a theorem.

(b) If there is an assignment of truth-values which makes $\sim B_1'$, $\sim B_2'$, ... simultaneously true, then there is an interpretation which satisfies $\sim A$, that is, A is not valid.

(c) But by (the law of infinite conjunction proved by) the infinity lemma, either for some k, $B_1' \vee B_2' \vee \ldots \vee B_k'$ is a tautology (i.e. $\sim B_1'$

$\& \sim B_2' \& \ldots \& \sim B_k'$ is a contradiction) <u>or</u>

there is an assignment which makes $\sim B_1', \sim B_2', \ldots$ simultaneously true.

Thus, to prove the theorem we need only show how to construct $B_1', B_2', \ldots$ and prove (a) and (b).

For, by (a)-(c), A is a theorem <u>or</u> A is not valid, i.e., A valid $\Rightarrow$ A theorem. Let the given formula A be

$$(Ey_1) \ldots (Ey_m)(z_1) \ldots (z_n)M[y_1, \ldots, y_m, z_1, \ldots, z_n]$$

where $y_1, \ldots, y_m, z_1, \ldots, z_n$ are all the variables of M.

<u>An ordering of m-tuples</u>

We order <u>all</u> m-tuples of natural numbers as follows:

$$\langle i_1, \ldots, i_m \rangle$$

comes before

$$\langle j_1, \ldots, j_m \rangle$$

if

$$(1) \quad (i_1 + \ldots + i_m) < (j_1 + \ldots + j_m)$$

or

$$(2) \quad (i_1 + \ldots + i_m) = (j_1 + \ldots + j_m)$$

and

$$i_1 = j_1, \ldots, i_k = j_k, \qquad i_{k+1} < j_{k+1}$$

for some k . Example for $m = 3$:

$$\begin{array}{c}
\langle 0,0,0\rangle \\
\langle 0,0,1\rangle \\
\langle 0,1,0\rangle \\
\langle 1,0,0\rangle \\
\langle 0,0,2\rangle \\
\langle 0,1,1\rangle \\
\langle 0,2,0\rangle \\
\langle 1,0,1\rangle \\
\langle 1,1,0\rangle \\
\langle 2,0,0\rangle \\
\langle 0,0,3\rangle \\
\vdots
\end{array}$$

Let the kth such m-tuple be

$$\langle [k1], [k2], \ldots, [km]\rangle .$$

<u>(m+n)-tuples</u>

From the kth m-tuple we form an associated (m+n)-tuple:

$\langle [k1], [k2], \ldots, [km], (k-1)n+1, \ldots, kn \rangle$

$k = 1$	$m = 3$	0 0 0	1 2
2	$n = 2$	0 0 1	3 4
3		0 1 0	5 6
4		1 0 0	7 8
5		0 0 2	9 10
6		0 1 1	...

motivate by prefix of $\neg A$ interpretation in domain natl. nos.

<u>Definition of B_k, B'_k, C_k, D_k</u>

Let B_k be the result of substituting the new variables:

$$x_{[k1]}, x_{[k2]}, \ldots, x_{[km]}, x_{(k-1)n+1}, \ldots, x_{kn}$$

for $y_1, y_2, \ldots, y_m, z_1, \ldots, z_n$ in M. B_4 is $M[x_1, x_0, x_0, x_7, x_8]$.

Let B'_k be formed from B_k by replacing $E_i(\ldots)$ by $p_{E_i(\ldots)}$ uniformly. I.e., to each atomic formula assign a unique statement letter of the prop. calc.

Let C_k be $B_1 \vee \ldots \vee B_k$.

Let D_k be $(x_0)(x_1)\ldots(x_{kn})C_k$ i.e., the closure of C_k. Note that the variables substituted for the z's are new and distinct.

(a) <u>Lemma</u>: <u>For every k, $\vdash D_k \supset A$</u> .

Proof by induction on k .

Basis: $\underline{\vdash D_1 \supset A}$

(1) $(\vec{z})\ M[\vec{y};\ \vec{z}] \supset (\overrightarrow{\exists y})(\vec{z})\ M[\vec{y};\ z]$

<u>argument</u>:

$B(x) \supset (\exists x)\ B(x)$ — $(x)\overline{B}(x) \supset \overline{B}(x)$ and prop. calc. Axiom 4

$(\vec{z})M \supset (\exists y_m)(\vec{z})M$

$(\exists y_m)(\vec{z})M \supset (\exists y_{m-1})(\exists y_m)(\vec{z})M$

...

$(\exists y_2)\ \ldots\ (\exists y_m)(\vec{z})M \supset (\overrightarrow{\exists y})(\vec{z})M$

and $(A \supset B) \wedge (B \supset C) \supset A \supset C$ — prop. calc.

and M.P.

(2) $(\vec{z})M[x_0,\ \ldots,\ x_0;\ \vec{z}] \supset A$

<u>argument</u>:

$(\vec{z})M[y_1,\ \ldots,\ y_m;\ \vec{z}] \supset A$ — (1)

$(y_m)((\vec{z})M[y_1,\ \ldots,\ y_m;\ \vec{z}] \supset A)$ GEN

$(y_m)((\vec{z})M[y_1, \ldots, y_m; \vec{z}] \supset A)$

$\supset . (\vec{z})M[y_1, \ldots, y_{m-1}, x_0, \vec{z}] \supset A$ Ax. 4 (no x_0 quantifiers)

$(\vec{z})M[y_1, \ldots, y_{m-1}, x_0; \vec{z}] \supset A$ MP

do this m times.

(3) $(x_1) \ldots (x_n)M[x_0, \ldots, x_0; x_1, \ldots, x_n] \supset A$

change of bound variable n times.

(4) $(x_0)(x_1) \ldots (x_n)M[x_0, \ldots, x_0; x_1, \ldots, x_n] \supset A$

<u>argument</u>:

$(x_0)[(x_1) \ldots (x_n)M \supset A]$ GEN

$(x_0)(x_1) \ldots (x_n)M \supset (x_0)A$ Thm. 3 and MP $(x)(A \supset B) \supset ((x)A \supset (x)B)$

$(x_0)(x_1) \ldots (x_n)M \supset A$ by Thm. 9 $(x)A \equiv A$ and Replacement x not free in A .

<u>Induction Step</u>

Assume $\vdash D_{k-1} \supset A$ and show $\vdash D_k \supset A$.
Note that C_k is $(C_{k-1} \vee B_k)$; D_k is $(x_0) \ldots (x_{kn})$ $(C_{k-1} \vee B_k)$.

$\vdash D_k \supset (x_0) \ldots (x_{(k-1)n})(C_{k-1} \vee (x_{(k-1)n+1}) \ldots (x_{kn})B_k)$ — Axiom 5. Noting that $x_{(k-1)n+1} \ldots (x_{kn})$ are not free in C_{k-1}. (They were z-values, new at k^{th} m+n-tuple.)

$(x)(\sim A(x) \supset B(x)) \supset ((Ex) \sim A(x) \supset (Ex)\ B(x))$ — Theorem 4

$(x)(A(x) \vee B(x)) \supset ((x)A(x) \vee (Ex)\ B(x))$ — by taut. from Theorem 4.

$\vdash D_k \supset (x_0) \ldots (x_{(k-1)n})C_{k-1} \vee (Ex_0) \ldots (Ex_{(k-1)n})(x_{(k-1)n+1}) \ldots (x_{kn})B_k$ — by above schema.

$\vdash D_k \supset (D_{k-1} \vee A)$ — Alph. change bd. vbl. (m+n).

$\vdash D_k \supset A$ — Prop. calc. and hypothesis $D_k \supset (D_{k-1} \vee A) \wedge (\neg D_{k-1} \vee A)$ $D_k \supset A$

Now if $B_1' \vee \ldots \vee B_k'$ is a tautology, any instance of it is a theorem, hence $B_1 \vee \ldots \vee B_k$ is a theorem. But this is C_k. But then by GEN, $D_k = (x_0) \ldots (x_{kn})C_k$ is a theorem. Hence by the above, and MP, A is a theorem.

(b) Suppose there is some assignment of truth-values which makes $\sim B_1'$, $\sim B_2'$, ... simultaneously true.

From this (master) assignment we construct an interpretation (in the domain D of the natural numbers) which satisfies $\sim A$.

To the q-adic predicate letter F assign a q-place relation R_F as follows: If $^pF(x_{i_1}, \ldots, x_{i_q})$ receives true, or is unassigned, in the master assignment, put $\langle i_1, \ldots, i_q \rangle$ in R_F . Otherwise $\langle i_1, \ldots, i_q \rangle \notin R_F$. This interpretation makes $\sim A$ true, hence A false.

<u>Proof</u>: $\sim A$ is $(y_1) \ldots (y_m)(Ez_1) \ldots (Ez_n) \sim M$. Consider an arbitrary m-tuple of elements of D, say the kth m-tuple. We must show that there exist other elements of D such that $\sim M$. But the kth (m+n)-tuple gives us the other elements: $(k-1)n+1, \ldots, kn$. For $\sim B_k$ is true under the master assignment to the $^pF(x_i \ldots x_j)$, and the interpretation gives $F(i, \ldots, j)$ the same tv as $^pF(x_i \ldots x_j)$; hence, under the interpretation $\sim M[[k1], \ldots, [km], (k-1)n+1, \ldots, kn]$ is true.

<u>Corollary</u>: (Skolem-Löwenheim Theorem) If A is satisfiable, then it is satisfiable in a <u>denumerable</u> domain. [By soundness. $\sim A$ not satisfiable in denumerable domain $\Rightarrow \vdash A \Rightarrow \sim A$ not satisfiable in <u>any</u> domain.])

<u>Corollary</u>: (Herbrand Theorem) If A is a wff in SNF and if B_k (subst. k^{th} $(m+n)$-tuple) and $C_k = \bigvee_{i=1}^{k} B_i$ are as above, then A is a theorem iff there is some k such that C_k is an instance of a tautology.

Note that this yields a <u>proof procedure</u>.

Remarks on the Completeness Theorem

Alternative proofs and reasons for this choice.

Constructive. Gives the domain.

Applications.

Examples: Non-theorem

Theorem

Note why this does not yield a decision procedure.

Proof procedures.

Theorem-proving by computer.

non-SNF: Enumerations.

Consequent complications of proof.

non-PNF: Herbrand Theorem.

Decision tables

Reduction to monadic for $(Ey)(z_1) \ldots (z_n)$

Solvable prefix cases.

Reduction classes.

Interprovability of

$(x)(y)M$	$(Ex)(y)M$	$(Ex)(Ey)M$
$(y)(x)M$	$(Ey)(x)M$	$(Ey)(Ex)M$
		$(x)(Ey)M$
		$(y)(Ex)M$

Satisfiability in a Denumerable Domain D

Remark: By the Skolem Löwenheim Theorem, which followed as a corollary of the proof of the Gödel Completeness Theorem, a formula A is satisfiable iff it is satisfiable in some denumerable domain.

Therefore, we consider a way of attempting to satisfy a wff A in a denumerable domain: i.e., of trying to find an interpretation which makes A true. By the result on the Skolem Normal Form we need consider for provability only formulas of the form $(y_1) \ldots (Ey_m)(z_1) \ldots (z_n)M$, M q-free, hence for satisfiability only formulas of the form

$$(*) \qquad (y_1) \ldots (y_m)(z_1) \ldots (z_n)M, \quad M \text{ q-free} .$$

In order to satisfy (*) we must find an interpretation, i.e., a denumerable domain and an assignment of relations to the predicate letters of M, such that (*) is true under the interpretation.

Decision Tables

An example: consider the wff

$$(x)(Ey)M$$

where M is $\sim F(x, x) \,\&\, \sim F(x, y)$ and suppose we wished

to show that this is satisfiable in some denumerable domain. We must find some <u>relation</u> R_F to correspond to F.

The formula must be <u>true</u>, i.e., for <u>every</u> $a \in D$, there must be some $b \in D$ such that $\sim F(a, a) \,\&\, \sim F(a, b)$. We represent the problem by a Decision Table (Church): As heading we put first the individual variables, then the atomic formulas.

x	y	F(x, x)	F(x, y)
Now for any		$a \in D$, there must be some	b such that:
a	b	F(a, a)	F(a, b)
		0	0
		i.e., $\langle a, a\rangle \notin R_F$	$\langle a, b\rangle \notin R_F$

EXAMPLE

Satisfiability (in a denumerable domain D) of $(x)(Ey)\text{-}M$, where M is a quantifier-free matrix which contains at most the atomic formulas $F(x, x)$, $F(x, y)$, $F(y, x)$, and $F(y, y)$.

x	y	F(x, x)	F(x, y)	F(y, x)	F(y, y)
0	1	F(0, 0)	F(0, 1)	F(1, 0)	F(1, 1)
1	2	---			

Suppose M is $F(x, x) \vee F(x, y)$. Then $-M$ is $-F(x, x) \,\&\, -F(x, y)$. To satisfy $(x)(Ey)\text{-}M$, we must find <u>some</u> relation to correspond to F for which the formula is <u>true</u>.

We need a relation R such that for every element a D there is some element b D such that (a, a) R and (a, b) R .

Using the decision table we can find such a relation:

a	b	F(a, a)	F(a, b)
		0	0
0	1	0	0
1	2	0	0
...			

In this cast it is clear that the empty relation R will satisfy the formula. Hence, since the negation is satisfiable, the formula (Ex)(y)M is not a theorem.

SET THEORY

REFERENCES

Naive Set Theory

Bourbaki, I. I. Théorie des Ensembles, Fascicule de Résultats, Hermann, Paris, 1958.

*Cantor, Transfinite Numbers, Dover, undated. (Original German edition, 1895-97.)

Hausdorff, Set Theory, Chelsea, 1957. (Translation of 1937, third German edition).

Kamke, The Theory of Sets, Dover, 1950.

Kelley, General Topology, Chapter 0, van Nostrand, 1955.

Mendelson, Introduction.

Axiomatic Set Theory

*Halmos, Naive Set Theory, van Nostrand, 1960.

Kelley, Appendix.

Mendelson, Chapter 4.

Wang and McNaughton, Les Systèmes Axiomatiques de la Théorie des Ensembles, Gauthier-Villars, Paris, 1953.

NAIVE SET THEORY

A <u>set</u> is a collection of objects. Cantor: "A set is a collection into a whole of definite, well-distinguished objects of our intuition or of our thought."

The objects in the collection are called <u>elements</u> or <u>members</u> of the set. $x \in y$ for x is a member of y. $\sim(x \in y)$ is written $x \notin y$.

A set x is a <u>subset</u> of y if every member of x is also a member of y. $x \subseteq y$.

To give a set, list its members

$$x = \{0, 1, 2\}$$

$$x = \{0, 1, 2, \ldots\} \quad ; \text{ or}$$

use a defining property

$$y = \{x | A(x)\}$$

where $A(x)$ is a predicate with only x free

$$y = \{x | x \text{ is a prime number}\} .$$

<u>Unit set</u> or <u>singleton</u> is a set with one member. [3]

Importance of the distinction between member and subset. $x \in A \Leftrightarrow \{x\} \subseteq A$. Example:

The <u>empty set</u> is a set with no members. $\emptyset$ or $\{\ \}$. Note that $\emptyset$ is not the same as $\{\emptyset\}$. In fact, $\emptyset \in \{\emptyset\}$. Also, $\emptyset \subseteq \{\emptyset\}$.

<u>Set equality</u>

$$x = y \text{ iff } x \subseteq y \text{ and } y \subseteq x .$$

Or,

$$(z)(z \in x \equiv z \in y) .$$

<u>Union and intersection</u>

$$x \cup y = \{z | z \in x \vee z \in y\}$$
$$x \cap y = \{z | z \in x \wedge z \in y\}$$

x and y are <u>disjoint</u> iff $x \cap y = \emptyset$.

<u>Complements</u>

$$\sim A = \{x | x \notin A\}$$
$$X \sim A = \{y | y \in X \wedge y \notin A\}$$

<u>Theorem</u>: $\emptyset \subseteq A \cap B \subseteq A \subseteq A \cup B$

<u>Theorem</u>: Let $A \subseteq X$, $B \subseteq X$. Then $A \subseteq B$ iff $A \cap B = A$ iff $B = A \cup B$ iff $X \sim B \subseteq X \sim A$ iff $A \cap X \sim B = \emptyset$ iff $(X \sim A) \cup B = X$.

Theorem: Let A, B, C and X be sets. Then

(a) $X \sim (X \sim A) = A \cap X$

(b) (Commutative laws) $A \cup B = B \cup A$

$A \cap B = B \cap A$

(c) (Associative laws) $A \cup (B \cup C) = (A \cup B) \cup C$

$A \cap (B \cap C) = (A \cap B) \cap C$

(d) (Distributive laws)

$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ and

$A \cup (B \cap C)$

(e) (de Morgan's laws)

$X \sim (A \cup B) = (X \sim A) \cap (X \sim B)$ and

$X \sim (A \cap B) = (X \sim A) \cup (X \sim B)$

Proofs:

(d) $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

$$\begin{aligned} x \in A \cap (B \cup C) &\Leftrightarrow x \in A \ \& \ x \in (B \cup C) \\ &\Leftrightarrow x \in A \ \& \ (x \in B \vee x \in C) \\ &\Leftrightarrow x \in A \ \& \ x \in B \vee x \in A \ \& \ x \in C \\ &\Leftrightarrow x \in A \cap B \ \vee \ x \in A \cap C \\ &\Leftrightarrow x \in (A \cap B) \cup (A \cap C) \end{aligned}$$

The cardinal number of a set of n elements is n. The cardinal number of the empty set is 0.

INFINITE SETS

DEFINITION: Two sets A and B are <u>equinumerous</u> ($A \simeq B$) iff their elements can be put into one-one correspondence (i.e., there is a one-one function F with range B and domain A).

DEFINITION: A and B have the same <u>cardinal number</u> iff $A \simeq B$. If $A \simeq B' \subset B$ and not $B \simeq A' \subset A$ then the cardinal number of A is less than the cardinal number of B. $\bar{A}$ = card A = cardinal number of A.

DEFINITION. (Dedekind) A set is <u>infinite</u> if it is equinumerous with some proper subset of itself. Otherwise it is <u>finite</u>.

DEFINITION: A set is <u>finite</u> if it is empty or if it is equinumerous with the set $\{0, 1, 2, \ldots, n-1\}$ of all natural numbers less than positive integer n. Otherwise it is <u>infinite</u>.

REMARK: The two definitions are equivalent, but the proof of their equivalence requires the Axiom of Choice.

DEFINITION: A set is <u>denumerable</u> or <u>countably infinite</u> if it is equinumerous with the set of all natural numbers. A denumerable set is said to have cardinality $\aleph_0$.

DEFINITION: A set is <u>countable</u> if it is finite or countably infinite.

DEFINITION: A set is uncountable (nondenumerable, nonenumerable) if it is not countable.

THEOREM 1: (Cantor) The set of all rational numbers is countable.

PROOF: We can imagine to be written down in order of magnitude, first, all whole numbers, i.e., all numbers with denominator 1; then all fractions with denominator 2; then all fractions with denominator 3, etc. There arise in this manner the rows of numbers

$$
\begin{array}{ccccc}
1 & 2 & 3 & 4 & \cdots \\
\frac{1}{2} & \frac{2}{2} & \frac{3}{2} & \frac{4}{2} & \cdots \\
\frac{1}{3} & \frac{2}{3} & \frac{3}{3} & \frac{4}{3} & \cdots \\
\frac{1}{4} & \frac{2}{4} & \frac{3}{4} & \frac{4}{4} & \cdot
\end{array}
$$

If we write down the numbers in the order of succession indicated by the line drawn in (leaving out numbers which are equal to ones which have already occurred), then

every positive rational number certainly appears, and only once. The totality of these rational numbers is thus written as a sequence

$$1,\ 2,\ 1/2,\ 1/3,\ 3,\ 4,\ 3/2,\ 2/3,\ 1/4,\ \ldots .$$

(This is Cantor's first diagonal argument.) If we denote the above sequence by

$$r_1,\ r_2,\ r_3,\ \ldots$$

then

$$0,\ -r_1,\ r_1,\ -r_2,\ r_2,\ \ldots$$

is an enumeration of <u>all</u> the rational numbers.

DEFINITION: <u>Power set</u> The power set of a set x is the set of all subsets of x. $\mathcal{P}(x)$

$A \in P(x) \Leftrightarrow A \subseteq X \Leftrightarrow \{x \mid x \in A \Rightarrow x \in X\}$.

<u>Example</u>: The set $\{0,1\}$ has power set $\{\{\ \},\ \{0\},\ \{1\},\ \{0,1\}\}$. Note that for a finite set of n members the size (cardinal number) of the power set is 2^n .

DEFINITION: We have denoted by $\aleph_0$ the number of integers. It is natural to denote by $2^{\aleph_0}$ the size of the power set, that is, the number of subsets of integers.

THEOREM: $2^{\aleph_0} > \aleph_0$.

PROOF:

	integers				
subsets	1	2	3	4	...
S	0				
S_1	1	0	0	...	
S_2	0	1	1		1 if the integer is in the subset, 0 otherwise.
S_3	1	1	0		
$\vdots$					

Now, diagonalize -- construct a set not in the list.

$$x \in S: \Leftrightarrow x \notin S_x$$

THEOREM: The set of reals is uncountable.

PROOF: Same. List,

r_1	0	9	1	0	...
r_2	8	1	3	1	...
r_3					

then the diagonal can be modified to give a real which differs from the nth real in the nth place.

$$\text{Set } d_n = \begin{matrix} 2 & \text{if } d^{(n)} = 1 \\ 1 & \text{if } d^{(n)} \neq 1 \end{matrix}.$$

The principles Cantor employed had previously been used for arguments about <u>finite</u> sets. He was the first to extend them to infinite sets. His work met with some disapproval and distrust, but his arguments appeared sound.

But, in 1902, the theory of sets was challenged by the discovery by <u>Russell</u> of a <u>paradox</u>.

RUSSELL's PARADOX (1902)

With the notation of naive set theory we can write:

$$y = \{x | x \notin x\}$$

So y is the set of all <u>sets</u> which are not members of themselves. Is $y \in y$?

If <u>yes</u>, then $y \in y$, hence $y \in \{x | x \notin x\}$ hence $y \notin y$.
If <u>no</u>, then $y \notin y$, hence $y \notin \{x | x \notin x\}$ hence $y \in y$.

What's wrong?

AXIOMATIC SET THEORY

A result of the discovery of the paradoxes of naive set theory was an attempt to axiomatize set theory. Since it was clear that to rely on the intuitive notion led to paradox, the solution appeared to be to state precisely the axiomatic basis for the theory. The basic problem appeared to be that we cannot consider sets which are too big. There are several such axiomatizations which so far appear to be consistent (contradiction-free). The most important of them are the system Z-F of Zermelo and Fraenkel, and the system NGB of von Neumann, Gödel, and Bernays. By Gödel's second incompleteness theorem we know that no such system can be proved to be consistent (without using methods which are in some sense more powerful than those of set theory.)

Problems with sets which are too big.

So maybe we should start with very small sets (which we can understand), and build up slowly in ways that seem reasonable.

Axioms of Set Theory

The system Z-F below is due to Zermelo and Fraenkel. (The major alternative NBG (von Neumann, Bernays, and Gödel) is given in Mendelson. NBG distinguishes between

sets (which may be elements), and classes (which cannot be elements)).

Notice that all axioms after the first assert set existence.

We start with predicate calculus, and introduce ϵ as a new primitive symbol.

1. Axiom of Extension

$$x = y \supset (w)(x \epsilon w \supset y \epsilon w)$$

Compare the definition of equality $(z)(z \epsilon x \equiv z \epsilon y)$. A set is determined by its elements. That is, if two sets have the same members, then everything true of one is true of the other.

2. Axiom of Unordered Pairs

Given sets x and y, $\{x,y\}$ is a set:

$$(Ew)(z)[z \epsilon w \equiv (z = x \vee z = y)]$$

Note that as a special case $\{x\}$ exists.

3. Power Set Axiom

For any set x, the set of all subsets of x (the power set of x) exists.

$$(z)(Ey)(x)[x \epsilon y \equiv (w)(w \epsilon x \supset w \epsilon z)]$$

or

$$(z)(Ey)(x)[x \epsilon y \equiv x \subseteq z]$$

or

$$(z)(Ey)[y = P(z)]$$

4. Axiom of Unions

$$(z)(Ey)(x)[x \in y \equiv (Ew)(x \in w \ \& \ w \in z)]$$

or

$$(z)(Ey)[y = \bigcup_{w \in z} w]$$

5. Aussonderungsaxiom (Axiom of Specification or Subsets)

Given a set z and a predicate $A(x)$ (or ZF), not containing free y, there is a subset of z containing all and only those sets x such that $A(x)$ is true:

$$(x)(Ey)(x)(x \in y \equiv (x \in z \ \& \ A(x)))$$

Compare this with the naive notion

$$x = \{y \mid A(y)\}$$

See how this resolves the Russell paradox. This gives the null set.

6. Axiom of Infinity

There is a set which contains the empty set and which, for every member of x, contains also the unit set of x:

$$(Ez)[\emptyset \in z \ \& \ (x)(x \in z \supset \{x\} \in z)]$$

7. Axiom of Regularity (Fundierungsaxiom)

Any nonempty set x contains a set y which is a minimal element.

$$(Ey)(y \in x) \supset (Ey)(y \in x \ \& \ \sim(Ez)[z \in x \ \& \ z \in y])$$

8. Axiom of Substitution (or Replacement) (Ersetzungsaxiom)

If the domain of a 1 - 1 function is a set, so is the range.

If $A(u, v)$ is a function, i.e.,

$$(x)(y)(z)(w)([A(x, y) \ \& \ A(z, w)] \supset (x = z) \equiv (y = w))$$

then, if there is a set of all sets u such that $(Ev)A(u, v)$, then there is a set of all sets v such that $(Eu)A(u, v)$.

9. Axiom of Choice

If x is a set of non-empty disjoint elements, then the union of x has at least one subset u having one and only one element in common with each member of x .

$$(x)([(y)(z)([y \in x \ \& \ z \in x]$$
$$\supset [(Ew)w \in y \ \& \ \sim(Ew)(w \in y \ \& \ w \in z)])$$
$$\supset ((Eu)(y)y \in x \supset (Ev)(t)(t = v \equiv t \in u \ \& \ t \in y)])]$$

The Cartesian product of a non-empty family of non-empty sets is non-empty.

Axiom of Choice (AxCh)

If $\alpha \to A_\alpha = \emptyset$ is a function defined for all $\alpha \in x$, then there exists another function $f(\alpha)$ for $\alpha \in x$, and $f(\alpha) \in A_\alpha$.

This allows us to do an infinite amount of "choosing" even though we have no property which would define the choice function and allow us to use Replacement instead.

We used AxCh in the Completeness Proof:

$$(x)(\exists y)\ A(x, y)$$

then

$$(\exists f)(x)\ A(x, f(y)) .$$

The existence of the Skolem function f follows from the Axiom of Choice.

Alternative Formulations of the Axiom of Choice

1. The Cartesian product of a non-empty family of non-empty sets is non-empty.

2. Given a non-empty class K of disjoint non-empty sets there exists a function f with range K such that $f(x) \in x$ for all members x of K.

This is provable by induction for finite K.

A <u>choice function</u>. Intuitively, the function f selects one element of each member A of K.

3. <u>Well-ordering principle</u>. Every set can be well-ordered. A set is well-ordered if every non-empty subset has a least element.

4. <u>Zorn's lemma</u>. If X is a non-empty partially-ordered set such that every chain in X has an upper bound, then X contains a maximal element.

<u>Axiom of Choice</u>

Of the axioms of set theory, the AXIOM OF CHOICE (given a family K of disjoint non-empty set $\exists f$ such that $f(x) \in x$ for each x in K) has seemed always to be less intuitively obvious than the others. Its expression is more complex and does not seem reducible to more basic notions. It has not been obvious that it might not be either contradictory--or else perhaps derivable from the others.

In 1939, Gödel, in a paper in the Proceedings of the National Academy of Sciences, followed in 1940 by an orange-covered publication, entitled, "The Consistency of the Axiom of Choice and of the Generalized Continuum Hypothesis with the Axioms of Set Theory," generally known as "The Monograph," proved that <u>if</u> the other axioms of set theory are consistent,

then set theory remains consistent if the Axiom of Choice and the Generalized Continuum Hypothesis are added.

Then, in 1962 63, Paul J. Cohen of the Mathematics Department at Stanford University proved another equally important and interesting result. The Axiom of Choice is in fact <u>independent</u>. That is, the axioms of set theory, if consistent, remain so, even if we assume that the axiom of choice is false. This shows, of course, that the Axiom of Choice is <u>not</u> a consequence of the other axioms. Furthermore, the continuum hypothesis is independent from the Axiom of Choice.

The proof of these results is beyond the scope of this course. See Paul J. Cohen, <u>Set Theory and the Continuum Hypothesis</u>, W. A. Benjamin, Inc., New York. 1966.

Let c = power of the continuum = $2^{\aleph_0}$.

<u>Continuum Hypothesis</u>

There is no A such that

$$\aleph_0 < \bar{\bar{A}} < 2^{\aleph_0} .$$

<u>Gödel</u>, 1939. Relative consistency of (AxCh) and GCH.

<u>Cohen</u>, 1963. Independence of AxCh and GCH and of GCH from AxCh. Cohen believes GCH is <u>false</u>.

PROOFS: by constructing <u>models</u> for the axioms of set theory which satisfy AxCh, GCH (Gödel) violate AxCh, GCH (Cohen).

REFERENCES ON TURING MACHINES

BOOKS

Mendelson, Chapter 5, Section 2, pp. 229-232.

Martin Davis, Computability and Unsolvability, McGraw-Hill, 1958.

B. A. Trakhtenbrot, Algorithms and Automatic Computing Machines, Heath, 1963.

Hartley Rogers, Recursive Functions and Effective Computability, forthcoming, McGraw-Hill.

Martin Davis, The Undecidable, Raven Press, 1965.

Marvin Minsky, Computation: Finite and Infinite Machines, Prentice-Hall, 1967.

PAPERS

A. M. Turing, On computable numbers with an application to the Entscheidungsproblem, Proc. London Math. Soc., sec. 2, 42 (1936-7) 230-265. Correction ibid 43 (1937), 544-546. (Reprinted in Davis, The Undecidable.

E. L. Post, Finitary combinatory process-formulation 1, Jour. Symbolic Logic. 1(1936), 103-105. (Reprinted in Davis, The Undecidable.)

Hao Wang, A variant to Turing's theory of computing machines. J. ACM 4 (1957), 63-92.

M. L. Minsky, Size and structure of universal Turing Machines using tag systems. Recursive Function Theory, Proc. Symp. Pure Math. V, A. M. S. 1962 229-238.

J. R. Büchi, Turing machines and the Entscheidungsproblem, Math. Ann. 148 (1962), 201-213.

J. C. Shepherdson and H. E. Sturgis, Computability of recursive functions, J. ACM 10 (1963), 217-255.

Michael O. Rabin and Hao Wang, Words in the history of a Turing Machine with a fixed input, J. ACM. 10 (1963), 526-527.

ALGORITHMS

DECISION PROBLEM (Entscheidungsproblem): Find an effective method to determine for any wff $\mathfrak{A}$ of the first-order predicate calculus whether or not $\mathfrak{A}$ is a theorem.

Suppose we had a suspicion that this was impossible --that there was no effective way of doing this; that there was no effectively calculable function f which when applied to a number x representing $\mathfrak{A}$ would produce 1 if $\vdash \mathfrak{A}$, 0 otherwise. How could we prove this?

Notice first that effective or effectively calculable is a good intuitive notion:

1. Some processes are clearly effective. (Deduction theorem, truth-tables, etc.)
2. Of other functions we can certainly say that we don't know enough about them to tell.

But to prove that some function is not effectively calculable we must have a precise notion. Suppose we make some assumption in the form:

THESIS: Every effectively calculable function is _____.

or

Every effective process is _____.

where the blank is filled in by some precise notion. Then we would be able to prove--to everyone who accepted our thesis--that certain functions are not effectively calculable.

The proofs will all be relative to the thesis. And the thesis itself is not subject to proof since it involves an intuitive notion. We cannot prove the thesis. But we can give what is called "the evidence for the thesis".

Most people accept the thesis to be presented. Notice that even if you do not, the proofs which will follow are still proofs--but they must then be qualified as relative to the thesis.

The THESIS has a number of versions. Perhaps the strongest evidence for it is that they are all equivalent, even though they have arisen under quite varied circumstances.

TURING'S THESIS: EVERY EFFECTIVELY CALCULABLE FUNCTION IS COMPUTABLE BY A TURING MACHINE.

CHURCH'S THESIS: EVERY EFFECTIVELY CALCULABLE (partial) FUNCTION IS GENERAL (partial) RECURSIVE.

MARKOV'S NORMALIZATION PRINCIPLE: Every algorithm in an alphabet A is fully equivalent relative to A to some normal (Markov) algorithm over A.

Similar, and equivalent, theses can be stated for Post normal systems, and for λ-definability.

The evidence for Church's thesis (Kleene):

(A) Heuristic evidence

(A1) Every particular effectively calculable function, and every operation for defining a function effectively from other functions, for which the question has been investigated, has proved to be general recursive.

(A2) The methods for showing effectively calculable functions to be general recursive are developed to a degree that it is impossible to imagine any effective process for evaluating a function which could not be transformed by these methods into a general recursive definition of the function.

(A3) Every attempt to get a function outside the class of general recursive function has either (1) not lead outside, or (2) given a function which is not effectively calculable.

(B) Equivalence of diverse formulations

(B1) As discussed above.

(B2) Stability of each of the notions. The several formulations of each of the main notions are equivalent. (To be shown for Turing machines.)

(C) The direct formulation of Turing machines from that of effective process.

Let us first consider informally the criteria which we would expect of an effective procedure. We look at the notion of algorithm, an effective process which always terminates.

An algorithm is a clerical (i.e., deterministic book-keeping) procedure which can be applied to any of a certain class of symbolic inputs, and which will eventually yield, for each such input, a corresponding symbolic output. We limit ourselves here to algorithms which take as input integers (or k-tuples of integers), and which output integers.

Well-known examples of algorithms are:

The sieve method for finding the n-th prime number.

The Euclidean algorithm for finding the greatest common denominator of x and y.

The following are some essential features of the informal notion of algorithm (see Rogers):

*1. An algorithm is given as a finite set P of instructions.

*2. There is a computing agent L, frequently human, which reacts to the instructions and carries out the computation.

*3. There are facilities for making, storing, and retrieving steps in a computation.

*4. The agent L reacts to the instructions of P in a discrete stepwise fashion, without using continuous methods or analog devices.

*5. The computation is carried forward deterministically --there are no random elements to be considered.

It is clear that the notion described contains a strong analogy to the description which could be made of any computation carried out by a digital computer. The notion of Turing machines dates back to 1936.

In addition to the criteria 1-5 above, there are other possible requirements which we might impose on the notion of algorithm. These requirements concern bounds on space and time. For example, we might (but do not) require the following:

6. A fixed bound on the size of inputs.

7. A fixed bound on the size of the set of instructions.

8. A fixed bound on the amount of storage space available.

9. A fixed bound on the length of the computation.

However, because it is possible to show that many functions which one would generally agree can be computed by algorithms cannot be computed within these restrictions 6-9, these are <u>not</u> to be taken as part of our informal definition.

By accepting one or more of 6-9, one can define interesting subclasses of functions and machines. These are being increasingly studied.

Even without 6-9, the notion above does place strong limitations on the capacity and ability of the computing agent. The agent can be restricted to

i. Clerical operations such as

read a symbol

move one symbol at a time backward or forward in the computation

move backward or forward through the instructions

write a symbol

ii. Fixed finite short term memory

iii. Fixed finite set of simple rules determining the operation to perform, and the next state of the short term memory.

We now describe the Turing machine, and will claim (Turing's thesis) that it formalizes the above notion.

Definition of a Turing Machine

Informally

A Turing machine carries out its operations on a two-way potentially infinite tape which is divided into squares:

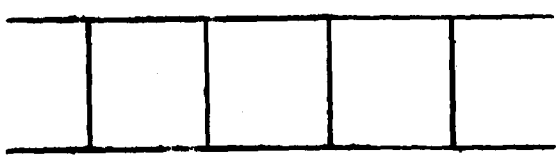


Fig. 1 (Davis)

By potentially infinite we mean that although the tape is at any time finite in length, additional squares can always be added at either the right or left-hand end of the tape.

There is a finite set of tape symbols $S_0, S_1, \ldots, S_n$ called the alphabet of the machine. (Turing argues convincingly that a finite set must suffice, since if the set were infinite, there would have to be symbols which differed by arbitrarily small amounts of printers' ink, were thus "arbitrarily close" and hence indistinguishable.)

The machine has a finite set of internal states $q_1, \ldots, q_m$. And at any given moment the machine is said to be in one of these states. Finally, there is a reading and writing head which at any moment stands over (scans) some square of the tape.

The machine just described acts only at discrete moments of time. And it is very limited in the acts it can perform.

If at a time t, the reading head is scanning a square containing a symbol S_i and the machine is in state q_j the next action, if any, of the machine is completely determined by an instruction set and must be one of the following:

1. Erase S_i and print a new symbol and change state.
2. Move one square left, change state.
3. Move one square right, change state.
4. Stop.

These actions can be represented by quadruples (following Post, rather than Turing, who used quintuples):

1. $q_j S_i S_k q_r$ (i may equal k)

2. $q_j S_i L q_r$

3. $q_j S_i R q_r$

4. absence of any quadruple beginning $q_j S_i$

The symbol S_0 is taken to represent the blank. Thus the machine always scans some symbol.

The Turing machine accepts as input a marked tape and begins in state q_1 scanning the left most symbol. The output is taken to be the contents of the tape at the time, if any, when the machine stops.

Formally:

DEFINITION: A Turing machine T is a finite set of quadruples of the above 3 kinds -- such that no two quadruples have the same first two symbols (deterministic)

DEFINITION: The alphabet of T is the set of tape symbols S_i which appear in the quadruples of T.
S_0 = blank = B

DEFINITION: The internal states of T are the symbols q_j which occur in the quadruples of T. q_1 is taken to be the initial state.

DEFINITION: An instantaneous description (complete configuration) of T is a word such that

i. All symbols but one are tape symbols of T

ii. One state symbol q_j occurs in the description, but is <u>not</u> the last symbol of the description.

DEFINITION: T moves one instantaneous description α into another β; $\alpha \underset{T}{\rightarrow} \beta$ if

α is	β is	and among the quads is
Pq_iS_jQ	$Pq_\ell S_kQ$	$q_iS_jS_kq_\ell$
$Pq_iS_jS_kQ$	$PS_jq_\ell S_kQ$	$q_iS_jRq_\ell$
Pq_iS_j	$PS_jq_\ell S_0$	
$PS_kq_iS_jQ$	$Pq_\ell S_kS_jQ$	$q_iS_jLq_\ell$
q_iS_jQ	$q_\ell S_0S_jQ$	

DEFINITION: T <u>halts</u> at an instantaneous description α iff there is no β such that $\alpha \underset{T}{\rightarrow} \beta$.

DEFINITION: A <u>computation</u> of T is a finite sequence of instantaneous descriptions $\alpha_0, \alpha_1, \ldots, \alpha_m$ such that q_1 is the left-most symbol of α_0, $\alpha_i \underset{T}{\rightarrow} \alpha_{i+1}$ for $0 \leq i < m$ and T halts on α_m.

Representation of integers

Let S_1 be 1 .

$\bar{m} = 1^{m+1}$ for any $m \geq 0$.

Let S_2 be * . Let $f(x_1, \ldots, x_m)$ be a function.

T <u>computes</u> f iff with input $\bar{k}_1 * \bar{k}_2 * \ldots * \bar{k}_m$ T halts only on $R_1 q_j R_2$ with $R_1 R_2 = Q$ and Q is = $R_3 \overline{f(k_1, k_2, \ldots, k_m)} R_4$ with R_3, R_4 possibly empty words consisting of S_0's only. (Mendelson)

T <u>computes</u> f iff with input as above T halts only on α_m where $(\alpha_m) = f(k_1, \ldots, k_m)$, where for any expression M, (M) is the number of occurrences of 1 in M . (Davis)

Examples

<u>Successor function $f(x) = x + 1$</u>

<u>Mendelson</u>	$q_1 1 L q_2$	<u>Davis</u>	$q_1 BBq_2$
	$q\ S_0 1 q_3$		or none

Example

$$q_1 1 L q_2$$

$$q_2 S_0 1 q_1$$

keeps on adding 1 to the left whenever the initial word starts with 1

Invariance of Turing Machines

We have given a particular definition of Turing machine in which we have specified that

the instructions are quadruples of a certain form,

the tape is 2-way infinite,

there is erasing (i.e., we can overprint any symbol with S_0) and there may be any finite number of symbols, and

there is only one tape.

Each of these conditions is inessential.

THEOREM: A TM with instructions which are quintuples can compute precisely the same functions as one whose instructions are quadruples.

Comment: The two formulations are not equivalent in all senses since, for example, with quintuples a universal Turing machine can be constructed with only two states (Shannon), though not with just one (Shannon); whereas with quadruples at least three states are required (Aanderaa). However, the differences do not effect the class of functions computed, but all concern measures for minimal machines.

PROOF:

1. Quints to quads

Replace each $q_i S_j S_k L q_m$ by $q_i S_j S_k q_i'$

$q_i' S_k L q_m$

where q_i' is a new state. Similarly for

$q_i S_j S_k R q_m$.

2. <u>Quads to quints</u>

Replace each $q_i S_j L q_m$ by $q_i S_j S_j L q_m$

$q_i S_j R q_m$ by $q_i S_j S_j R q_m$.

But q ints must move, so replace each

$q_i S_j S_k q_m$ by $q_i S_j S_k L q_i'$

q_i' new and add the ℓ instructions

$q_i' S_\ell S_\ell R q_m$ for all S_ℓ .

NOTE: Advantage is fewer instructions.

<u>Example</u>: $f(x) = 2x$ <u>Using Davis' convention</u>

q_1	1	B	R	q_2	erase extra 1
q_2	1	1'	R	q_3	mark 1 to 1' to indicate "copied"
q_3	1	1	R	q_3	move R to first B and write 1"
q_3	1"	1"	R	q_3	
q_3	B	1"	R	q_4	
q_4	B	B	L	q_4	move L until 1 or 1' is encountered: go to q_5 or q_6
q_4	1"	1"	L	q_4	
q_4	1'	1'	R	q_6	
q_4	1	1	L	q_5	
q_5	1	1	L	q_5	move to left-most 1
q_5	1'	1'	R	q_2	
q_6	1"	1	R	q_6	now have $\underbrace{1'\ 1'\ \ldots\ 1'}_{x}\ \underbrace{1''\ 1''\ \ldots\ 1''}$
q_6	B	B	L	q_7	change 1" 's to 1's
q_7	1	1	L	q_7	go L to 1' 's and change them to 1's
q_7	1'	1	L	q_7	
q_7	B	———			no instruction

Under Mendelson's convention we would skip first instruction and use q_1 for q_2 .

Example: $f(x) = 2x$ in **quadruples**

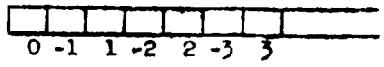
q_1	1	B	q_1'	q_1'	B	R	q_2
q_2	1	1'	q_2'	q_2'	1'	R	q_3
q_3	1	R	q_3				
q_3	1"	R	q_3				
q_3	B	1"	q_3'	q_3'	1"	R	q_4
q_4	B	L	q_4				
q_4	1"	L	q_4				
q_4	1'	R	q_6				
q_4	1	L	q_5				
q_5	1	L	q_5				
q_5	1'	R	q_2				
q_6	1"	1	q_6'	q_6'	1	R	q_6
q_6	B	L	q_7				
q_7	1	L	q_7				
q_7	1'	1	q_7'	q_7'	1	L	q_7

THEOREM: A one-way infinite tape suffices.

Comments: Was used by Turing.

Frequently more useful in applications; the decision problem proof uses them.

PROOF: Fold the tape,



and modify the program.

THEOREM: (Wang) Erasing is dispensable.

<u>Comment</u>: Therefore, computers could get by with paper tape.

THEOREM: Two-symbols suffice.

PROOF: Use a suitable encoding.

bab, baab, baaab, ...

<u>or</u> use $\log_2 n$ squares.

THEOREM: Programmed Turing machines will do. (Wang).

$\leftarrow$	R
$\rightarrow$	L
α	write α
$T(\alpha, n)$	conditional transfer.

To be done in detail later (SS).

THEOREM: Two-states suffice. (Shannon)

THEOREM: Triples suffice (exercise).

Various extensions: n-tapes; n-dimensional tapes; jumps; etc.

A numbering of Turing machines

Since each Turing machine is formally a set of quadruples, it is possible to assign numbers to them so that we may refer to the n-th Turing machine. For example, we might use Gödel numbers:

Suppose we assign to each of the symbols which may occur in a quadruple of some Turing machine a distinct odd number ≥ 3 . Viz:

3	5	7	9	11	13	15	17	. .
R	L	S_0	q_1	S_1	q_2	S_2	q_3	. .

Then the Gödel number (gn) of a quadruple is

$$2^a 3^b 5^c 7^d$$

where a, b, c, d are the gn of the 4 symbols.

Example:

$$gn(q_1 S_1 R q_2) = 2^9 3^{11} 5^3 7^{13} .$$

Further, with a sequence $M_1, M_2, \ldots, M_n$ of quadruples, we associate the gn:

$$2^{gn(M_1)}3^{gn(M_2)} \ldots Pr(n)^{gn(M_n)}$$

where $Pr(n)$ = n-th prime.

The numbering here is not unique since we have not specified an order for the quadruples. Thus, each TM has $n!$ gn, where n is the number of quadruples. But given these gn we can find a unique gn for each Turing machine by simply taking the smallest of the $n!$ numbers.

<u>Fundamental Theorem of Arithmetic</u> (For proof see Appendix to Davis):

Every integer $x > 1$ can be represented in the form $p_1^{m_1}p_2^{m_2} \ldots p_k^{m_k}$ where the p_i are unique primes. Moreover, this representation is unique except for the order of the factors. By the Fundamental Theorem, no two of the gn which we have produced are the same.

This gives a mapping of TM into the integers. Note that given any number we can tell whether or not it is the gn of a TM. From these numbers we can then obtain an <u>onto</u> mapping by assigning 1 to the TM given by the smallest such gn, 2 to the next (unless some permutation of the quads has already been counted), and so on. We shall thus speak of the n-th TM assuming that some such serial numbering has been adopted.

Universal Turing Machines

Each Turing machine appears to correspond to a special-purpose digital computer. One of the main results of Turing's paper was the description of the Universal Turing Machine, which in a sense corresponds to a general-purpose machine. The UTM, given a suitably encoded version of an arbitrary Turing machine T and an input n, produces the same output as T does with input n .

Note that the gn of T could be used as the encoding of T.

Minimal Turing machines

Turing machines can be classified in complexity by the state-symbol product (a measure introduced by Shannon). The following problem then arises: what is the minimal state-symbol product for a UTM. The current best solution is due to John Cocke and Marvin Minsky, who have shown that 4 states and 7 symbols suffice.

If we allow more than 1 tape, the result can be improved. Hooper (1963) proved that 2 states, 3 symbols, and 2 tapes suffice; likewise that 1 state, 2 symbols, and 4 tapes suffice, even requiring that one of the tapes be a fixed loop.

BLANK PAGE

Read: Trakhtenbrot

THE HALTING PROBLEM

We are now in a position to present an <u>unsolvable problem</u>. Note that an unsolvable problem is actually an unsolvable class of problems.

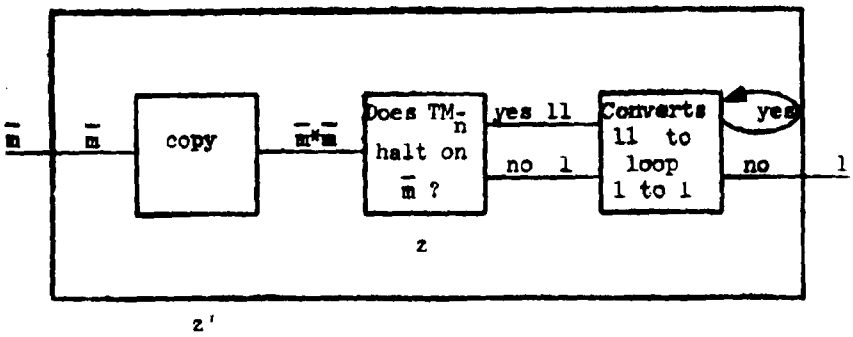
HP: Given the number n of a TM and an input m, does TM n halt for input m ?

We suppose that there exists such a machine and derive a contradiction. Let TM_z be the machine which solves the problem. That is, TM_z computes the function φ_z:

$$\varphi_z(m,n) = \begin{cases} 1 \text{ if } TM_n \text{ halts on } m \\ 0 \text{ otherwise} \end{cases}$$

We show that given TM_z we can effectively construct a $TM_{z'}$ which is self-contradictory.

1. We can construct a TM which from input $\tilde{m}$ produces output $\tilde{m}*\tilde{m}$.
2. We construct a TM which on input 11 loops, and on input 1 halts with output 1 .
3. We then <u>compose</u> (effectively) these two machines with TM_z to produce $TM_{z'}$ which has the desired property.



$$\varphi_z'(k) = \begin{cases} 1 & \text{if } TM_k \text{ does not halt on } k; \text{ i.e., } \varphi_k(k) \text{ is undefined.} \\ \text{loops} & \text{if } TM_k \text{ halts on } k, \text{ i.e., } \varphi_k(k) \text{ is defined.} \end{cases}$$

$$\varphi_z'(z') = \begin{cases} 1 & \text{if } \varphi_z'(z') \text{ undefined} \\ \text{undefined} & \varphi_z'(z') \text{ defined} \end{cases}$$

<u>Converter</u>

$$\left.\begin{array}{l} q_1\ 1\ R\ q_2 \\ q_2\ 1\ L\ q_1 \end{array}\right\} \begin{array}{l} \text{on input } 11 \text{ loops} \\ \text{on input } 1 \text{ halts with output } 1 \end{array}$$

<u>copying machine</u>

input $\bar{m}$
output $\bar{m}*\bar{m}$

[Note that this is a minor variation of the machine previously given to compute 2n .]

$$\left.\begin{array}{l} q_1\ 1\ R\ q_1 \\ q_1\ B\ *\ q_2 \end{array}\right\} \text{go right and print } *$$

$$\left.\begin{array}{l} q_2\ *\ L\ q_2 \\ q_2\ 1\ 1'\ q_3 \\ q_2\ 1'\ L\ q_2 \\ q_2\ B\ R\ q_5 \end{array}\right\} \text{go L to 1 put 1'}$$

(continued on next page)

q_3	1'	R	q_3	go R to B and print 1
q_3	*	R	q_3	
q_3	1	R	q_3	
q_3	B	1	q_4	
q_4	1	L	q_4	go L to *
q_4	*	L	q_2	
q_5	1'	1	q_5	tidy up
q_5	1	R	q_5	
q_5	*	L	q_6	
q_6	1	L	q_6	go back to start.
q_6	B	R	q_7	

Composition

It is only fair to note that the construction of $TM_{z'}$ in detail requires a proof that any two TM can be composed. I.e., that a new TM can be obtained by using the output of the first machine as the input to the second. This can be carried out formally the details are given in Davis. What is involved is showing that we can assume that the second machine begins in the required initial form, that is, scanning the left-most symbol of the output of the first machine. This can be proved by the use of <u>n-regular</u> machines, which always terminate with instantaneous description $q_k \ldots$ where q_k is a state symbol for which there are <u>no</u> instructions in the first machine.

Effectiveness

Note that the construction of $TM_{z'}$ from TM_z is effective--it could be done by a TM.

Reductions of the halting problem (HP)

General method: Given problem P reduce HP to P .
I.e., show if we could solve P this would give a solution to HP . Conclude cannot solve P .

Covering:

$\underline{a}$	$\underline{b}$
$\{a_i\}$	$\{b_i\}$
a_i	$f(a_i)$

If from a solution to $f(a_i)$ one can derive a solution to a_i then $\{b_i\}$ covers $\{a_i\}$.

$\therefore$ if $\underline{a}$ unsolvable, so is $\underline{b}$.

Example:

WP for semigroups.

Example:

Decision Problem for 1st-order predicate calculus.

Example:

HPB (see below).

⋮

Example:

Printing Problem

Example:

Blank tape - Is the TM tape ever blank?
(for later use)

THEOREM: THE HALTING PROBLEM FOR TM WITH BLANK TAPE IS UNSOLVABLE.

There is no algorithm (no Turing machine) which will decide, given as input the numeral $\bar{n}$ whether or not the n-th TM halts when started with a blank tape.

PROOF:

REMARK: The proof is of a form which is standard in proofs of unsolvability. We use the reduction of a known unsolvable problem to the new problem. That is, we show that a solution to the new problem would yield a solution to a problem which has no solution. Therefore, the new problem is unsolvable.

This proof is therefore important both for the result, which we will need, and as a simple example of a very important method.

NOTE: Need first to reduce HP to HP for single numeral as input. Proved above.

LEMMA: For any numeral $\bar{m}$, we can effectively construct a TM which starts with blank tape, and halts with the instantaneous configuration $q_k\bar{m}$, where the states of the TM are only $q_0, q_1, \ldots, q_k$ and there are no instructions beginning q_kS_i for any S_i.

PROOF OF LEMMA:

REMARK: Note that we do not claim that there is **some** TM which will work for any $\bar{m}$, but only that for any $\bar{m}$ there is a TM .

$q_0 B1Rq_1$ $\qquad$ $q_{m+1} 11Lq_{m+1}$

$q_1 B1Rq_2$ $\qquad$ $q_{m+1} BBRq_{m+2}$

$\vdots$

$q_{m-1} B1Rq_m$

$q_m B1Lq_{m+1}$

PROOF OF THE THEOREM:

Suppose the halting problem for TM with blank tape could be solved. We can then solve HP as follows:

To decide if TM_n halts on input $\bar{m}$, form a new TM by changing each quintuple of TM_n as follows:

$$q_i S_j S_k \left\{\begin{smallmatrix} L \\ R \end{smallmatrix}\right\} q_l \quad \text{to} \quad q_{i+m+2} S_j S_k \left\{\begin{smallmatrix} L \\ R \end{smallmatrix}\right\} q_{l+m+2}$$

and adding the above instructions for the TM which writes $\bar{m}$ on blank tape. Then the new TM will halt on blank tape iff TM_n halts on input $\bar{m}$.

Q.E.D.

REMARK: Note that in this construction, as in those which will follow, we are using the formulation of TM in terms of quintuples rather than quadruples. We have previously given a proof that they are equivalent.

BLANK PAGE

PARTIAL RECURSIVE FUNCTIONS

Church's thesis: Every partially computable function is partial recursive. (Extended form.)

A partial recursive function is really a recursive partial function. Partial vs. total.

Definition of partial recursive function

A function is partial recursive if it can be obtained from the initial functions of schemata I, II and III below by a finite number of applications of schemata IV, V and VI.

$$\left.\begin{array}{ll} \text{I.} & S(x_1) = x_1 + 1 \\ \text{II.} & \mathcal{O}^n(x_1, \ldots, x_n) = 0 \\ \text{III.} & U_i^n(x_1, \ldots, x_n) = x_i \end{array}\right\} \text{Initial functions}$$

IV. Composition If $h, g_1, \ldots, g_m$ are partial recursive, so is the function f defined by

$$f(x_1, \ldots, x_n) = h(g_1(x_1, \ldots, x_n), \ldots, g_m(x_1, \ldots, x_n))$$

V. Primitive recursion If g, h are partial recursive, so is the function defined by

$$f(0, x_2, \ldots, x_n) = g(x_2, \ldots, x_n)$$

$$f(z + 1, x_2, \ldots, x_n) = h(z, f(z, x_2, \ldots, x_n), x_2, \ldots, x_n)$$

VI. <u>Minimalization</u> If g is partial recursive, so is the function f defined by

$$f(x_1, \ldots, x_n) = \mu_y[g(x_1, \ldots, x_n, y) = 0]$$

"μ_y" is "the least y such that"
$f(x_1, \ldots, x_n)$ is defined to be y_0 iff

$$g(x_1, \ldots, x_n, y_0) = 0 \text{ and } (\forall y < y_0)[g(x_1, \ldots, x_n, y)$$

is defined and non-zero].

DEFINITION: A partial recursive function is <u>general recursive</u> (or total) if it can be defined by I - VI in such a way that in all applications of VI,

$$(x_1) \ldots (x_n)(Ey)(g(x_1, \ldots, x_n, y) = 0) .$$

DEFINITION: A (general) recursive function is <u>primitive recursive</u> (PR) iff it can be defined without use of schema VI.

Example for primitive recursive:

$$[f_1(x, y) = x + y]$$

y^x: $f_1(0, y) = U_1^1(y)$

$f_1(x', y) = S(U_2^3(x, f_1(x, y), y))$

(we write x' for $x + 1$)

$$[f_2(x, y) = x \cdot y]$$

$f_2(0, y) = \mathbb{O}^2(0, y) = 0$

$f_2(x', y) = f_1(U_2^3(x, f_2(x, y), y), U_3^3(x, f_2(x, y), y))$

$f_3(0, y) = S(\mathbb{O}^2(y)) = 1$

$f_3(x', y) = f_2(U_2^3(x, f_3(x, y), y), U_3^3(x, f_3(x, y), y))$

<u>Not all re... functions (even of 1 varisble) are primitive recursive</u>.

As in halting problem proof we diagonalize:

1. We can gödel number the PR functions of 1-vbl.:

   gödel number the symbols (introducing ;),

   then the expressions; can effectively decide

   if PR.

   Hence we can talk of the xth PR function of

   1-vbl.

2. Now diagonalize.

Let P_x be the xth PR fn.

Then $P_x(x) + 1$ is computable.

But it is not PR. For suppose

$f(x) = P_x(x) + 1 = P_e(x)$ for some e.

Then

$$f(e) = P_e(e) + 1 = P_e(e)$$

<u>contradiction</u>.

This argument would not go through

for partial recursive functions:

because could conclude only that $P_e(e)$ undefined;

for general recursive functions:

because we cannot effectively decide if general recursive (step 1 fails).

We now prove the equivalence of Turing's Thesis and Church's Thesis by showing first that all recursive functions are machine computable, then that all machine computable functions are partial recursive.

THEOREM: All partial recursive functions are machine computable. We shall prove this by giving a series of machines ending in the very simple SS-machine.

Reference: J. C. Shepherdson and H. E. Sturgis, Computability of Recursive Functions, JACM, Vol. 10, No. 2, April 1963, pp. 217-255.

Also, preliminary version of above:
J. C. Shepherdson, The computability of partial recursive functions by forms of Turing machines. (mimeographed.)

The URM (Unlimited Register Machine)

Infinity of registers $\boxed{1}\boxed{2}\boxed{3}\boxed{\ldots}$ each of which can store any natural number 0, 1, 2, . . .

Denote by $\langle n \rangle$ the contents of nth register.

Instructions:

$P(n)$: $\langle n \rangle = \langle n \rangle + 1$

$D(n)$: $\langle n \rangle = \langle n \rangle - 1$ if $\langle n \rangle \neq 0$

$\Theta(n)$: $\langle n \rangle = 0$

$C(m,n)$: $\langle n \rangle = \langle m \rangle$

$J[k]$: Unconditional transfer to line k of program

$J(m)[k]$: Transfer to line k if $\langle m \rangle = 0$

This is a very powerful machine; it is therefore easy to show that every recursive function can be computed. [Compare proof in Kleene which works directly with Turing Machines.]

DEFINITION: We say that a partial recursive function f of n arguments is URM-computed if it is computed by the URM in the following sense: For every set of natural numbers $x_1, x_2, \ldots, x_n, y, N$ $(y \neq x_i, x_i, y \leq N$ for $1 \leq i \leq n)$ there exists a routine $R_N(y = f(x_1, \ldots, x_n))$ such that if $\langle x_1 \rangle, \ldots, \langle x_n \rangle$ are the initial contents of registers $x_1, \ldots, x_n$, then if $f(\langle x_1 \rangle, \ldots, \langle x_n \rangle)$ is undefined the machine will not stop, if $f(\langle x_1 \rangle, \ldots, \langle x_n \rangle)$ the machine will stop with $\langle y \rangle = f(\langle x_1 \rangle, \ldots, \langle x_n \rangle)$ and with contents of all registers $1, 2, \ldots, N$ (<u>except</u> y) the same as their initial contents.

THEOREM: Every partial recursive function can be URM-computed.

PROOF:

I. $R_N(y = S(x))$

1. $C(x, y)$
2. $P(y)$

II. $R_N(y = \Theta(x))$

1. $\Theta(y)$

III. $R_N(y = U_i^n(x_1, \ldots, x_n))$

1. $C(x_i, y)$

IV. <u>Composition</u>

$R_N(y = f(x_1, \ldots, x_n))$ where f defined by IV

1. $R_{N+1}(N + 1 = g_1(x_1, \ldots, x_m))$

2.

⋮

m. $R_{N+m}(N + m = g_m(x_1, \ldots, x_n))$

m+1. $R_{N+m}(y = h(N + 1, \ldots, N + m))$

V. <u>Primitive Recursion</u>

Notation: Let I be a subroutine. Then $I^{\langle n \rangle}$ goes through I $\langle n \rangle$ times and sets $\langle n \rangle = 0$.

(I must have single normal exit.)

$I^{\langle n \rangle}$: 1. $J(n)[2]$, I, $D(n)$, $J[1]$

2.

$R_N(y = f(x_1, \ldots, x_n))$ where f defined by V

1. $R_N(y, g(x_2, \ldots, x_n))$, $\Theta(N + 1)$

2. $\{R_{N+2}(N + 2 = h(N + 1, \langle x_1 \rangle, y, x_2, \ldots, x_n))$
$C(N + 2, y)$, $P(N + 1)\}$

restores register x_1 3. $C(N + 1, x_1)$

VI. <u>Minimalization</u>

$R_N(y = f(x_1, \ldots, x_n))$ where f is defined by VI

1. $\Theta(y)$

2. $R_{N+1}(N + 1 = g(x_1, \ldots, x_n, y))$

3. $J(N + 1)[4]$, $P(y)$, $J[2]$

4.

This will loop if $\mu_y[\ \ldots\ ]$ is undefined.

$\therefore$ We have for each partial function f a subroutine $R_N(y = f(x_1, \ldots, x_n))$ which URM-computes it.

$\boxed{X}$

The convention regarding subscript N for subroutines can be extended to instructions: We write

$$P_N(n),\ D_N(n),\ O_N(n),\ C_N(m, n),\ J_N[k],\ J_N(m)[k]\ .$$

<u>Reduction of instruction set</u>

The large (6) instruction set of the URM was convenient in the above proof. But we can eliminate three of them:

$O_N(n)$:

1. $J_N(n)[4]$
2. $D(n)$
3. $J[1]$
4.

$C_N(m,n)$: 1. $O_N(n)$, $O_{N+1}(N+1)$

2. $\{P_{N+1}(N+1), P_{N+1}(n)\}^{\langle m \rangle}$

3. $[P_{N+1}(m)]^{\langle N+1 \rangle}$

Now $\overline{J}_N(m)[k]$ can be added and used to eliminate first $J_N(m)[k]$ and then $J_N[k]$. $\overline{J}_N(m)[k]$ is transfer on nonzero to k

$J_N(m)[k]$: 1. $\overline{J}_N(m)[2]$, $J_N[k]$.

2.

$J_N[k]$: 1. $P_{N+1}(N+1)$, $\overline{J}_{N+1}(N+1)[k]$

Thus we have only the instructions

$P_N(m)$

$D_N(m)$

$\overline{J}_N(m)[k]$

where subscript N indicates that registers beyond N may be used as workspace and may be altered, but that registers 1 through N are preserved.

We move toward our very restricted final machine by now introducing the LRM, Limited Register Machine. The LRM has three instructions above, but no longer has an infinite number of registers. It has a potentially infinite number, the actual number being controlled by the two additional instructions

$N \to N + 1$

$N \to N - 1$ remove an empty register.

We remove a register (empty or not) by the subroutine

$N \to_1 N - 1$: 1. $P_N(N)$

2. $D_N(N)$, $\bar{J}_N(N)[2]$

3. $N \to N - 1$.

THEOREM: All partial recursive functions are LRM-computable.

PROOF

Take the URM which computes the function and find the maximum instruction subscript M . Replace all subscripts by M . If $M > N$, add initially the instructions $N \to N + 1$, $N + 1 \to N + 2$, ..., $N + M - 1 \to N + M$ and add at end $N + M \to_1 N + M - 1$, ..., $N + 1 \to_1 N$.

<u>SS-machines</u>

The SS-machine is a one-register machine with alphabet {', *} and three instruction types: F^0, F^1 and $SCD[m_1, m_2]$. F^0 and F^1 are <u>write</u> instructions, which print $a_0 = '$ or $a_1 = *$ at the right end of the register. $SCD[m_1, m_2]$ is a <u>scan and delete</u> of the leftmost symbol, which operates as follows:

If no leftmost symbol, take next instruction.

If leftmost is a_0, delete and go to m_1 .

If leftmost is a_1, delete and go to m_2 .

THEOREM: Every partial recursive function is computable by the SS-machine.

PROOF: (By reducing the LRM to a single register machine with these instructions.)

The storage medium of the LRM at any time consists of the contents of N registers:

$$\langle 1\rangle, \langle 2\rangle, \ldots, \langle N\rangle$$

Introduce the new symbol ' and think of memory as a single word:

$$\langle 1\rangle \text{ ' } \langle 2\rangle \text{ ' } \ldots \text{ ' } \langle N\rangle$$

LEMMA: There is a subroutine T which will change the word A_1 ' A_2 ' ... ' A_N into A_2 ' ... ' A_N ' A_1 .

PROOF:

T: 1. P^0, SCD[3, 2]

2. P^1, SCD[3, 2]

3. ---

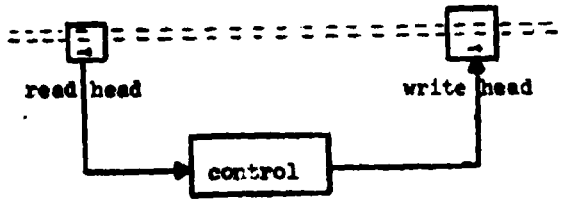
Let T^n be $T, \ldots, T$ (n times). Obtain the LRM operations by bringing the word to be operated on to the beginning, operating on it, and restoring it to its original position.

$P_N(n)$	T^n, P^1, T^{N-n}
$D_N(n)$	1. T^{n-1}, SCD[2, 2] 2. T^{N-n+1}
$J_N(n)[k]$	1. T^{n-1}, P^0, SCD[2, 3] 2. T^{N-n}, SCD[k, k] 3. P^1, SCD[4, 3] 4. T^{N-n}
$\bar{J}_N(n)[k]$	1. $J_N(n)[3]$ 2. P^0, T^N, SCD[k, k] 3. ---
$N \to N+1$	P^0
$N \to N-1$	1. T^{N-1}, SCD[2, 2] 2.

Remark: SCD[m_1, m_2] can be further weakened to SCD(m): jump on 1, proceed to next if 0 .

Physical realizations of SS-machine

I.



each head moves after it acts

II. Deck of cards

(1) Add card marked , at top.

(2) Add card marked * at top.

(3) Draw bottom card. If , go to m_1.

If * go to m_2.

If empty go to next instruction.

While {', *} suffices, we could also define an s+1 symbol SS-machine.

THEOREM: Any SS-computable function is Turing computable.

PROOF:

From SS M we construct TM Z which has symbols s_0 and s_1 and also the blank symbol S_0. Corresponding to each instruction of m there is a state of Z:

m	q_m
1	q_1

Corresponding to the configuration of M at instruction m with tape contents ⟨tape⟩, we have Z in the configuration $S_0 q_m \langle tape \rangle S_0$.

Corresponding to each instruction of M, a set of instructions of Z:

m: P^i

$$\begin{cases} \text{go right to first blank} \\ \text{print } a_i \\ \text{go left to first blank} \\ \text{go right } 1 \end{cases}$$

$$\begin{cases} q_m a_j R q_m & j = 0, 1 \\ q_m S_0 a_i q'_m & \\ q'_m a_j L q'_m & j = 0, 1 \\ q'_m S_0 R q_{m+1} & \end{cases}$$

m: $SCD[m_0, m_1]$

$$\begin{array}{ll} q_m a_j S_0 q'_{m_j} & j = 0, 1 \\ \underline{q_m S_0 \; S_0 q''_m} & \\ q'_{m_j} S_0 R q_{m_j} & j = 0, 1 \\ q''_m S_0 R q_{m+1} & \end{array}$$

COROLLARY: Every partial recursive function is Turing-computable.

The second half of the equivalence of Turing's thesis and Church's thesis is given by the following theorem.

THEOREM: Every (partial) function computable by a Turing machine is (partial) recursive.

Reference: Davis, Chapter $\left\{\begin{matrix}\text{III, Section 5}\\ \text{IV, Sections 1 and 2}\end{matrix}\right\}$

Informal Sketch of Proof:

<u>Gödel numbers</u>

We have assigned gn to TM .

Review:

3	5	7	9	11	
↕	↕	↕	↕	↕	...
R	L	S_0	q_1	S_1	

gn of an expression

$$a_1 \cdots a_n = \prod_{k=1}^{n} P_n(k)^{gn(a_k)}$$

gn of a sequence of expressions

$$M_1, \ldots, M_n = \prod_{1}^{n} P_n(k)^{gn(M_k)} .$$

Note: The power of 2 in the gn of an expression is <u>odd</u>. In a sequence it is <u>even</u>.

We gave before the special case of expressions which were TM .

Kleene's T-predicate

We define a predicate $T_n(z, x_1, \ldots, x_n, y)$, i.e., $T_n(z, \vec{x}, y)$, which is to mean for given $z, x_1, \ldots, x_n$ and y that z is a gn of a Turing machine Z, and Y is the gn of a computation of Z beginning with the instantaneous description $q_1(\overline{x_1, \ldots, x_n})$.

DEFINITION: A predicate is $\left\{\begin{matrix}\text{primitive recursive}\\ \text{partial recursive}\end{matrix}\right\}$ according as its characteristic function is (true = 0, false = 1).

THEOREM 1: $T_n(z, \vec{x}, y)$ is primitive recursive.

Proof uses the fact that bounded minimalization, $\mu_{y \leq z}$, is primitive recursive.

U is a primitive recursive function such that if y is the gn of a computation, then $U(y)$ is the output of the computation.

THEOREM 2: Let Z_0 be a TM and z_0 a gn of Z_0 . Then the domain of the function $\varphi_{z_0}^{(n)}(\bar{x})$ is equal to the domain of $\mu y T_n(z_0, \vec{x}, y)$. Moreover

$$\varphi_{z_0}^{(n)}(\vec{x}) = U(\mu y T_n(z_0, \vec{x}, y)) .$$

KLEENE NORMAL FORM THEOREM

$f(\vec{x})$ is partially computable iff $\exists z_0$ such that:

$$f(\vec{x}) = U(\mu y T_n(z_0, \vec{x}, y)) .$$

Corollary: Every (partially) computable function is (partial) recursive.

[illegible]

DEFINITIONS [illegible]

[illegible]

DEFINITIONS

Normal [illegible]

Monogenic Matrices [illegible]

Test [illegible]

Best possible result [illegible]

THEOREM (Post)

There are monogenic normal systems with unsolvable halting problems.

PROOF (Wang)[1]

Take a Universal SS-machine with n instructions. Set up a corresponding monogenic normal system. Use alphabet of SS-machine plus $2(n+1)$ new symbols $b_1, \ldots, b_{n+1}$, $e_1, \ldots, e_{n+1}$.

(1) $0 \to 0$

(2) $1 \to 1$

For each instruction q_i which is P^0:

(3) $b_i \to b_{i+1}$

(4) $e_i \to 0e_{i+1}$

For each instruction q_i which is P^1

(5) $b_i \to b_{i+1}$

(6) $e_i \to 1e_{i+1}$

For each instruction q_i which is $SD(k)$

(7) $b_i 0 \to e_{i+1} b_{i+1}$

(8) $b_i 1 \to e_k b_k$

---

[1] NOTE: Can also do for $Sd(k, m)$, by adding $b_i e_i \to b_{i+1} e_{i+1}$.

(9) $e_i e_{i+1} \to e_{i+1}$

(10) $e_i e_k \to e_k$

Then the SS-machine halts on input $x_1 x_2 \ldots x_p$. This system halts on the starting word $b_i x_1 x_2 \ldots x_p e_i$.

PROOF

SS-machine	normal system
<u>start</u> $x_1 x_2 \ldots x_p$	$b_i x_1 x_2 \ldots x_p e_i$
<u>i^{th} instruction is P^0</u>	
$x_1 x_2 \ldots x_p 0$	$x_1 x_2 \ldots x_p e_i b_{i+1}$
	$\vdots$
	$e_i b_{i+1} x_1 \ldots x_p$
	$b_{i+1} x_1 \ldots x_p 0 e_{i+1}$
<u>i^{th} instruction P^1</u>	
similarly	
<u>i^{th} instruction is SD(k)</u>	
$0x_2 \ldots x_p$	$b_i 0x_2 \ldots x_p e_i$
$\downarrow$	
	$x_2 \ldots x_p e_i e_{i+1} b_{i+1}$
$x_2 \ldots x_p$	$\vdots$
	$e_i e_{i+1} b_{i+1} x_2 \ldots x_p$
	$b_{i+1} x_2 \ldots x_p e_{i+1}$
$1x_2 \ldots x_p$	$b_i 1x_2 \ldots x_p e_i$
$\downarrow$	$1x_2 \ldots x_p e_k b_k$
	$\vdots$
$x_2 \ldots x_p$	$e_k b_k x_2 \ldots x_p$
	$b_k x_2 \ldots x_p e_k$

POST CORRESPONDENCE PROBLEM

Emil L. Post, A variant of a recursively unsolvable problem, Bull. A.M.S., Vol. 52, No. 4 (April, 1946), pp. 264-268.

Correspondence problem:

To determine for an arbitrary finite set (g_1, g_1'), $\ldots$, (g_μ, g_μ') of pairs of corresponding non-null strings on a, b whether there exist $n \geq 1$, $i_1, \ldots, i_n$ such that

$$g_{i_1} g_{i_2} \cdots g_{i_n} = g_{i_1}' g'_{i_2} \cdots g_{i_n}' .$$

Examples

1) pairs: (b^3, b^2)

(ab^2, bab^3)

solution: $g_1 g_2 g_1 = b^3 ab^2 b^3 = b^2 bab^3 b^2 = g_1' g_2' g_1'$

2) pairs: (a, ba)

$(b^2 a, a^3)$

$(a^2 b, ba)$

solution: clearly no solution, since there is no pair to start with.

References in application to ALGOL

Cantor, JACM 9(62), pp. 477-479.

Floyd, CACM 5(62), p. 526, p. 534.

Post's proof of the unsolvability of the correspondence problem began with the unsolvability of the decision problem for the class of normal systems on a, b . He reduced the problem for normal systems to the correspondence problem, hence showing that the correspondence problem must be unsolvable.

We shall obtain the unsolvability of the correspondence problem by reducing the halting problem for SS-machines.

<u>Post Correspondence Problem</u>

LEMMA: If the SS-machine M computes the partial function $f(x_1, \ldots, x_n)$ then there is an SS-machine M' which

1. halts iff M halts;
2. never has an empty tape, except possibly at start and end.

PROOF:

Go back to the LRM. The function f can be LRM computed by a program which begins by adding a register $(N \to N + 1)$ and storing * in it. Ends by deleting * and $N \to N - 1$. The SS version of this program will then begin with the instruction P^0 .

LEMMA: If the SS-machine M' computes f as above, we can construct an SS-machine M" which 1. halts iff M' halts (hence if M halts) and 2. halts only on an empty tape.

PROOF: Construct M" from M' by replacing any halts by

$$L_h: \ SCD(L_h , L_h) .$$

M" of course does not compute a very interesting function, but it is defined for the same inputs as f .

LEMMA: The halting problem for SS-machines is unsolvable.

PROOF: By equivalence with TM .

LEMMA: The halting problem for SS-machines starting with blank tape is unsolvable.

PROOF: By equivalence with TM .

LEMMA: The following problem is unsolvable for SS-machines: Does SS-machine starting with blank tape ever get back to blank tape?

PROOF: By previous lemmas.

PCP (A modification of Dana Scott's proof.)

For any SS-machine with line $L_1 \ldots L_n$ effectively construct corresponding PCP.

Construction

(0e, e0)

(1e, e1)

(e, eL_1) for initial instruction

L_i: P^0		(L_ie, e0eL_{i+1})
L_i: P^1		(L_ie, e1eL_{i+1})
L_i: SCD[L_j, L_k]		(L_ie0e, eL_j)
all j, k		(L_ie1e, eL_k)
if j = k	add	(L_ieL_i, eL_i)
L_n: no instruction		no pairs

To prove: PCP has a solution iff M starting with blank tape gets back to blank tape.

PROOF: 1. Both words must begin with (e, eL_1)

2. Both words must end with (L_{n-1}eL_n, eL_n) for some L_n.

Example 1

L_1: $P^{(0)}$

L_2: $SCD[L_2, L_2]$

L_3: halt

(0e, e0)	(e, eL_1)	(L_1e, e0eL_2)
(1e, e1)		(L_2e0e, eL_2)
		(L_2e1e, eL_2)
		(L_2eL_3, eL_3)

$$\text{right}\begin{cases} \underline{e} \quad \underline{e} \quad \underline{0e} \quad \underline{L_2e0e} \quad \underline{L_2eL_3} \\ \underline{eL} \quad \underline{e0eL_2} \quad \underline{e0} \quad \underline{eL_2} \quad \underline{eL_3} \end{cases}.$$

Example 2

L_1: P^0

L_2: P^1

L_3: $SCD[L_2, L_3]$

L_4: halt

What does it do on blank tape?

L_1: 0

L_2: 01

L_3: 1

L_2: 11

L_3: 1

L_3: ^

L_4:

Pairs (0e, e0), (13, 31), (e, eL_1)

(L_1e, e0eL_2)

(L_2e, e1eL_3)

(L_3e0e, eL_2)

(L_3e1e, eL_3)

(L_3eL_4, eL_4)

e L_1e 0e L_2e 0e 1e L_3e0e 1e L_2e 1e 1e L_3e1e 1e

eL_1 e0eL_2 e0 e1eL_3 e0 e1 eL_2 e1e1eL_3 e1 e1 eL_3

L_3e1eL_3eL_4
L_2e1eL_2eL_4 ok

If L_i writes, it is followed by tape after L_i .

If L_i reads, it is followed by tape before L_i .

Example 3

1. P^0
2. Scd[6, 3]
3. P^1
4. P^1
5. Scd[6, 3]
6. Scd[6, 6]

Ambiguity Problem for Context-free Grammars

Phrase-structure grammars and rewriting systems.

$$(V, T, S, P) \qquad T \subset V$$
$$S \in V - T$$

Context-free grammar.

$$P: \; A \to \varphi \qquad A \in I$$
$$\varphi \text{ a string in } V$$

Example:

$$\{0e^{i_n}0e^{i_{n-1}}0 \ldots 0e^{i_1}0x_{i_1} \ldots x_{i_n} \mid n \geq 0\}$$

1. is CF
2. define ambiguity
3. is unambiguous

THEOREM: The ambiguity problem for CFG is unsolvable.

PROOF: By reduction of the PCP to the ambiguity problem.

References

Hao Wang, Proving Theorems by pattern recognition - II, Bell System Technical Journal, 40(1961), 1-42.

A. S. Kahr, Edward F. Moore, and Hao Wang, Entscheidungsproblem reduced to the AEA case, Proceedings of the National Academy of Sciences, U.S.A., 48(1962), 365-377.

Hao Wang, Dominoes and the AEA case of the decision problem, Proceedings of the Symposium on Mathematical Theory of Automata, Polytechnic Institute of Brooklyn, 1962, 23-55.

Robert Berger, The undecidability of the domino problem, doctoral thesis, Harvard University, 1964. Computation Laboratory Report No. BL-37.

THE DOMINO PROBLEM

The domino problem, introduced by Wang in reference 1, is an amusing combinatorial problem which can be very simply stated and which has some important consequences.

STATEMENT OF DOMINO PROBLEM

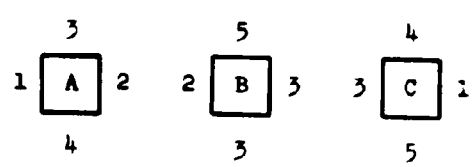
A <u>domino set</u> is a finite set of types of square plates, the <u>dominoes</u>, all of the same size, whose edges are marked with symbols (or <u>colors</u>), each plate in a different manner. There are an infinite number of copies (also called <u>dominoes</u>) of each type.

The infinite plane is assumed to be ruled into domino-size squares, and we seek to assemble the dominoes onto the plane according to the rules:

1. No domino may be reflected or rotated.
2. A domino must be placed exactly over a square.
3. The symbols on adjacent domino edges must match.
4. Every square must be covered with a domino.

A domino set is said to be solvable if we can cover the entire plane in this way.

EXAMPLE



We can obtain a solution to this set by using the block

A	B	C
C	A	B
B	C	A

Which has on the periphery the symbols

	3	5	4	
1				1
3				3
2				2
	3	5	4	

Since the top edge of the 3x3 block is the same as the bottom edge, and the left edge the same as the right edge, we can repeat this block in every direction to cover the entire plane.

<u>The domino problem</u> is the following general problem:

Is there an algorithm (a decision procedure) by which given an arbitrary domino set P, we can decide whether P is solvable?

Berger, 1964. NO.

DEFINITION: A <u>torus</u> of a domino set is a rectangle of dominos such that

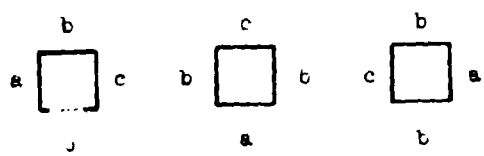
1. adjacent edges have the same color
2. the bottom edge is the same as the top edge
3. the left edge is the same as the right edge.

THEOREM: Every set which has a torus is solvable.

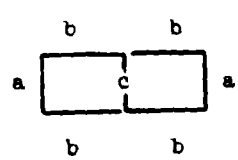
PROOF: We can cover the entire plane with the torus.

DEFINITION: A solution of a domino set is periodic if there is a torus T such that the solution can be viewed as made up entirely of copies of T.

<u>Example</u>:



Note that the example has a torus



and therefore has a periodic solution. The torus, and hence the periodic solution, used only two of the three dominoes of the set.

REMARKS:

The definition of <u>periodic</u> does not include all solutions which might possibly be considered to be in some sense periodic, but is arbitrarily restricted.

<u>Example</u>:

		1				3				4				3				4	
1		A		1	1		B		2	2		C		2	1		D		4	4		E		4
		1				1				1				3				4	

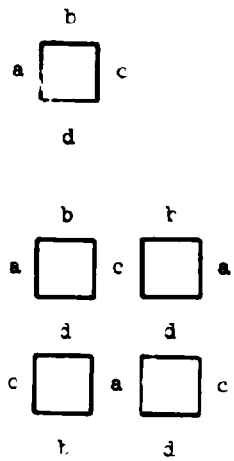
<u>Solutions:</u>

(a) all A (b) all E (c) A $\begin{array}{c|c} D & E \\ \hline B & C \end{array}$ A

(d) $\begin{array}{c} E \\ \hline C \\ \hline A \end{array}$ (e) A|B|E

THEOREM: If rotations and reflections were allowed, the problem would be trivial, i.e. every set would have a (periodic) solution.

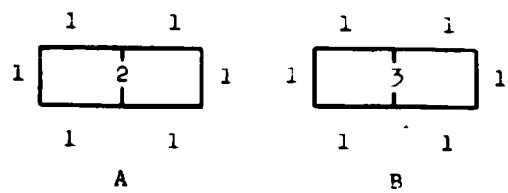
PROOF:



THEOREM: A set may have both periodic and nonperiodic solutions.

Pf:

Set P_1:

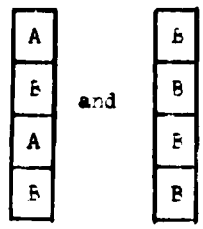


With either torus we get periodic solutions. Using both we can obtain as many different solutions as there are binary infinite sequences (i.e. $2^{\aleph_0}$).

PROOF:

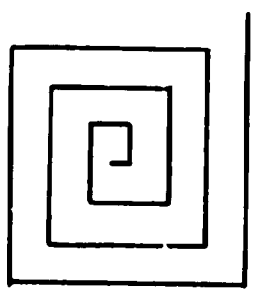
Number (the rectangles of squares of) the plane around the origin. To fix the origin assign

And then use



to build the infinite number of solutions. The solutions cannot be translated into one another since

occurs only at the origin.



Questions on periodic solvability:

DOES EVERY SOLVABLE SET HAVE A TORUS? (No. Berger 1964)

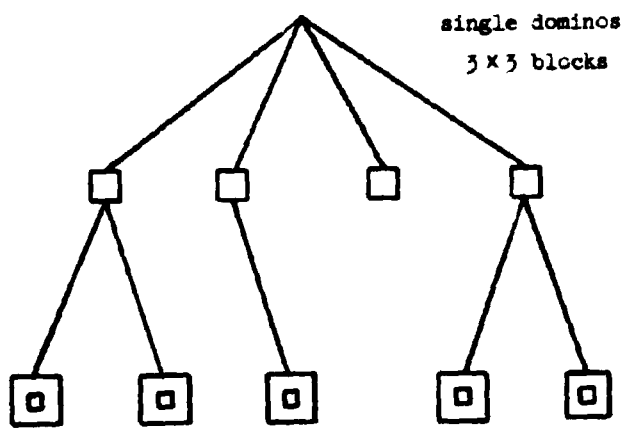
Subquestion: DOES SOME SOLVABLE SET HAVE SOME SOLUTION THAT CONTAINS NO TORUS? (BERGER using Thue, yes. But cannot eliminate the periodic solutions.)

THEOREM: (Berger) There exists a domino set which has a solution which contains no torus.

THEOREM: A set is solvable on the whole plane iff it is solvable in a quadrant.

PROOF: ⇐ trivial

⇒



Infinite number of levels. Qed by infinity lemma. Note that this is non-constructive - it does not enable us to <u>find</u> a solution.

BLANK PAGE

CONSTRAINED DOMINO PROBLEMS

So far we have considered only the unconstrained or general domino problem. That is, given a domino set P, can the plane be filled with the dominoes of P.

One might also consider domino problems which are in some way constrained:

The origin-constrained problem:

Given a set $D = P \cup Q$, can we fill the plane subject to the restriction that the origin is filled with a domino of P.

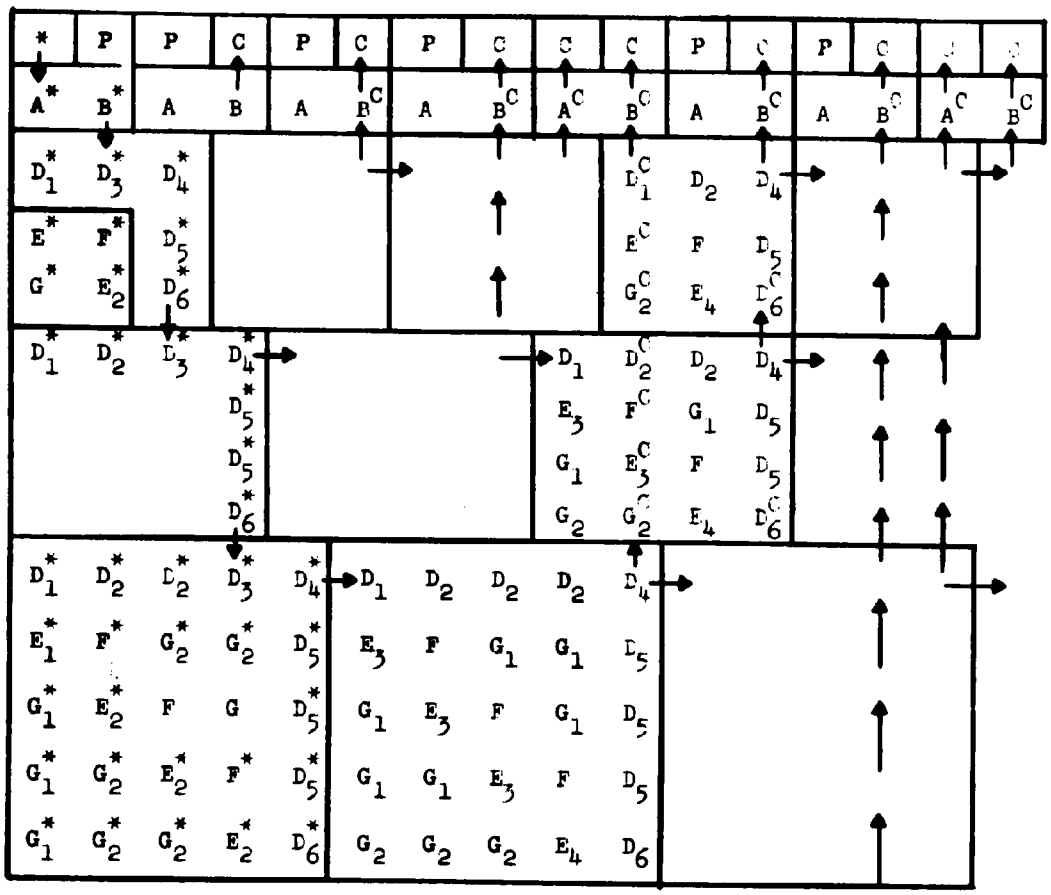
Computation by dominoes

If we consider solutions in an infinite quadrant and require a fixed domino * to occur at the origin, we can usually find domino sets with unique solutions satisfying a variety of given conditions.

An example of a puzzle which can be solved with dominoes is the following.

> Find a set of dominoes such that if * is required to appear at the origin, it has a unique solution in which P and C occur respectively at the prime and composite squares in the first row.

The following solution uses 38 dominoes and includes some improvements due to M. Fieldhouse of a solution initially obtained by E. F. Moore and Hao Wang. (Smaller solutions are possible). The form of the solution is indicated in the diagram below:



The arrows indicate important signals. The * indicates that the domino is affected by the initial boundary condition, and the superscript C indicates that the domino is transmitting the 'number is composite' signal (except that D_6^C is absorbing this signal). The dominoes B (which is used only once) and D_4 generate the 'number is composite' signal.

We assume the left-hand margin is color 0, and the top color 0. The solution is unique. The 38 dominoes may be defined as follows:

0 / 0 $*$ 1 / 51		0 / 1 P 1 / 1		0 / 1 C 1 / 21	
51 / 0 A^* 51 / 32	1 / 51 B^* 2 / 52	1 / 2 A 1 / 26	21 / 1 B 2 / 26	21 / 2 A^C 1 / 226	21 / 1 B^C 2 / 226
32 / 0 D_1^* 3 / 37	32 / 3 D_2^* 3 / 32	52 / 3 D_3^* 53 / 32	26 / 53 D_4^* 30 / 53	53 / 2 D_5^* 2 / 53	53 / 7 D_6^* 2 / 52
26 / 30 D_1 3 / 7	26 / 3 D_2 3 / 24		226 / 3 D_4 30 / 3	3 / 2 D_5 2 / 3	3 / 77 D_6 2 / 26
226 / 30 D_1^C 3 / 27	226 / 3 D_2^C 3 / 224		32 / 7 F^* 2 / 37		3 / 77 D_6^C 2 / 226
37 / 0 E_1^* 7 / 32	37 / 2 E_2^* 7 / 32	7 / 2 E_3 7 / 24	24 / 7 F 2 / 7	24 / 2 G_1 2 / 24	24 / 2 G_2 2 / 26
32 / 0 G_1^* 2 / 32	27 / 2 E_4^C 77 / 226	27 / 2 E_3^C 7 / 224	224 / 7 F^C 2 / 27	224 / 2 G_1^C 2 / 224	224 / 2 G_2^C 2 / 226
32 / 2 G_2^* 2 / 32	7 / 2 E_4 77 / 26				

THEOREM: The origin-constrained domino problem is unsolvable.

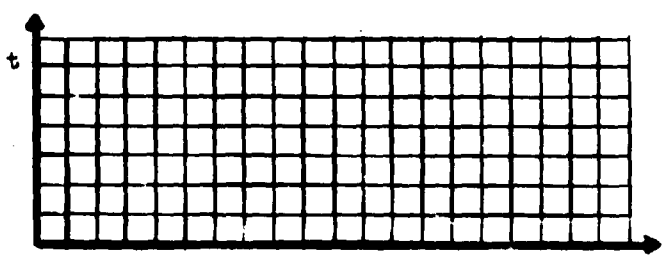
PROOF: Recall: The halting problem for TM starting with blank tape is unsolvable.

Recall: A TM can be restricted to a one-way infinite tape.

We give a <u>general</u> <u>method</u> which when applied to <u>any</u> Turning Machine X produces a corresponding set of dominoes P_x, with a distinguished type D such that:

X halts on an initially blank tape $\Leftrightarrow$ P has no solution with D at the origin.

Plot TM configurations in the plane:



We illustrate the method by applying it to a specific machine X:

$$q_1S_0S_1Rq_2 \qquad q_1S_1S_1Rq_1$$

$$q_2S_0S_0Rq_3 \qquad q_2S_1S_1Lq_3$$

$$q_3S_0S_1Lq_4 \qquad q_3S_1S_0Lq_4$$

$$q_4S_0S_0Lq_1$$

<u>P_X consists of the following domino types:</u>

A. Two types for each tape symbol

$$[S_0], \quad [LS_0], \quad [S_1], \quad [LS_1]$$

B. One type for each <u>permissible</u> kind of scanned square (state and symbol):

$$[q_iS_j] \quad \begin{matrix} i = 1,\ldots, 4 \\ j = 0,1 \end{matrix} \qquad (i,j) \neq (4,1)$$

C. One type for the next scanned square (symbol and next state) after a left-shift

$$[Lq_iS_j] \quad i = 1,3,4 \quad j = 0,1$$

D. One type for the next scanned square after a right shift

$$[Rq_i S_j] \quad i = 1,2,3 \quad j = 0,1$$

E. Four types for the initial row and column

[D] for origin

[B] for beginning of tape

[↑] for initial row

[→] for initial column

Machine X will halt at step B . We want to color the dominoes so that the only possible solution is the partial solution below:

t									
9	→		S_0						
8	→	q_4S_1 L	q_3S_1	S_1	S_0	S_0			
7	→	LS_1	q_3S_1 L	q_2S_1	S_0				
6	→	LS_1	q_1S_0	q_2S_1 R	S_0				
5	→	q_1S_1	q_1S_0 R	S_1	S_0				
4	→	q_1S_1 L	q_4S_0	S_1	S_0				
3	→	LS_1	q_4S_0 L	q_3S_0	S_0				
2	→	LS_1	q_2S_0	q_3S_0 R	S_0	S_0			
1	→	q_1S_0	q_2S_0 R	S_0	S_0	S_0	S_0	S_0	...
0	D	B	↑	↑	↑	↑	↑	↑	↑

Figure 1

Let us now describe this domino problem as a prelude to coloring the dominoes:

The conditions on P_x

1. origin constraint

$$[D]00 \qquad (\pounds x)[D]^{**}$$

2. the initial row and initial column are the boundary

2.1 $[D]xy \supset [B]x'y$

2.2 $([B] \vee [\uparrow]xy) \supset [\uparrow]x'y$

2.3 $([D] \vee [\rightarrow]yx') \supset [\rightarrow]yx'$

3. the next row above the initial row simulates the initial configuration

3.1 $[B]yx \supset [q_1S_0]yx'$

3.2 $[\uparrow]yx \supset ([Rq_2S_0] \vee [S_0])yx'$

4. the left or right neighbor of the scanned square at time y is in part determined by a left or right shift and embodies information for the scanned square at time y'.

Notation: $[Lq_i]$ for $([L_{qi}S_0] \vee [L_{qi}S_1])$

$[Rq_i]$ for $([Rq_iS_0] \vee [Rq_iS_1])$

4.1 $[q_i S_j]x'y \supset [Lq_k]xy$

for $(i,j,k) = (2,1,3),(3,0,4),(3,1,4),(4,0,1)$

4.2 $[q_i S_j]xy \supset [Rq_k]x'y$

for $(i,j,k) = (1,0,2),(1,1,1),(2,0,3)$

5. <u>the state and scanned square at time y' are determined by $[Lq_i]$ or $[Rq_i]$ at time y.</u>

5.1 $[Lq_i S_j]yx \supset [q_i S_j]yx'$ $\quad i = 1,3,4 \quad j = 0,1$

5.2 $[Rq_i S_j]yx \supset [q_i S_j]yx'$ $\quad i = 1,2,3 \quad j = 0,1$

6. <u>the tape symbol at time y' and position x is determined by the tape symbol at (x,y)</u>

6.1 $[S_i]yx \supset ([S_i] \vee [Rq_1 S_i] \vee [Rq_2 S_i] \vee [Rq_3 S_i])yx'$

for $i = 0,1$

$[LS_i]yx \supset ([LS_i] \vee [Lq_1 S_i] \vee [Lq_3 S_i] \vee [Lq_4 S_i])yx'$

for $i = 0,1$

6.2 $[q_2 S_0]yx \supset [Lq_4 S_0]yx'$

$[q_3 S_1]yx \supset [S_0]yx'$

$[q_4 S_0]yx \supset ([Rq_1 S_0] \vee [Rq_2 S_0])yx'$

$[q_1 S_0]yx \supset ([LS_1] \vee [Lq_3 S_1])yx'$

$[q_1 S_1]yx \supset [LS_1]yx'$

$[q_2 S_1]yx \supset [S_1]yx'$

$[q_3 S_0]yx \supset (S_1]yx'$

7. <u>in each row $[S_i]$ and $[LS_i]$ must be distinguished</u>

7.1 $([Rq_1] \vee [Rq_2] \vee [Rq_3] \vee [q_2S_1] \vee [q_3S_0] \vee [q_3S_1]$

$\vee [q_4S_0] \vee [S_0] \vee [S_1])xy \supset ([S_0] \vee [S_1])x'y$

7.2 $([Lq_1] \vee [Lq_3] \vee [Lq_4] \vee [q_1S_0] \vee [q_1S_1] \vee [q_2S_0]$

$\vee (LS_0) \vee [LS_1])x'y \supset ([LS_0] \vee [LS_1] \vee [\rightarrow])xy$

8. <u>the halting conditions</u>

8.1 $\neg[q_4S_1]xy$

we have gotten 8.1 by simply excluding the type

8.2 $[\rightarrow]xy \supset \neg([Lq_1] \vee [Lq_3] \vee [Lq_4])xy$

this could be deleted if we had included the condition that no two types can be assigned to the same place

<u>Argument</u>: These conditions are sufficient to determine the colors on the domino types.

We have not excluded the case in which several dominoes occur at the same place. (1 assures that at least one occurs). To express this condition we could add an explicit condition:

9. only one type at xy

How to color the dominoes. Each domino gets four colors. We give the colors somewhat unusual names - as shown in Figure 2.

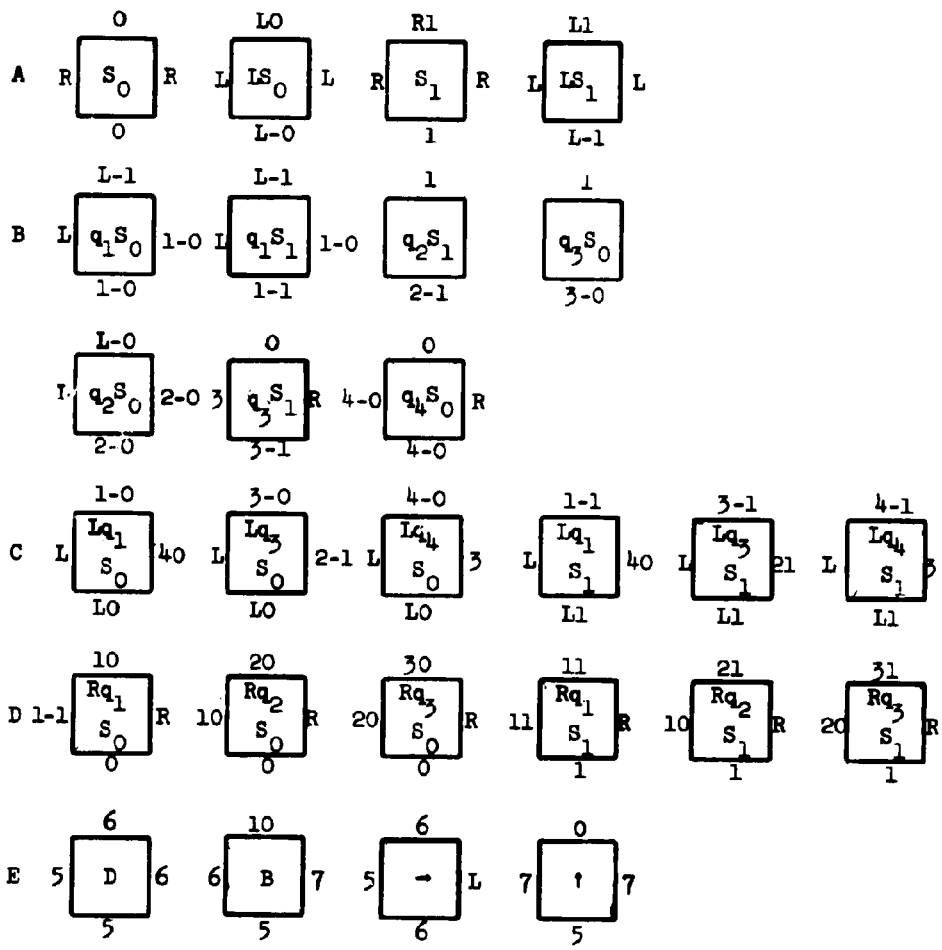


Figure 2

Let $\mathcal{D} = \{D_1, \ldots, D_n\}$ be a finite set of domino types.

Let $F^i xy = F^i(x,y)$ be square (x,y) is covered by a domino of type D_i.

Let $\mathcal{D}_0 = \{D_1, \ldots, D_k\}$ $k \leq n$ be a subset of $\mathcal{D}$ to be used at the origin.

Let $R_i = \{j \mid 1 \leq j \leq n \wedge D_j$ has on left the same color D_i has on right$\}$.

Let $T_i = \{j \mid 1 \leq j \leq n \wedge D_j$ has on bottom the same color D_i has on top$\}$.

Then we can fill the quadrant with dominoes of the types in $\mathcal{D}$ iff the following conditions are met:

<u>Every square has precisely one domino type</u>:

$$\bigvee_{i = 1,\ldots, n} (F^i xy \wedge \bigwedge_{\substack{j = 1,\ldots, n \\ j \neq i}} \neg F^j xy)$$

<u>The domino to the right matches correctly</u>:

$$F^i xy \supset \bigvee_{j \in R_i} F^j x'y$$

<u>The domino on top matches correctly</u>:

$$F^i yx \supset \bigvee_{j \in T_i} F^j yx'$$

And the origin constraint can be expressed as:

$$\bigvee_{1 \leq i \leq k} F^{i}00$$

Büchi's lemma

A formula $EzKz \wedge \forall x \exists u \forall y Mxuy$ in which K and M are quantifier-free is satisfiable iff $K0 \wedge \forall x \forall y Mxx'y$ is satisfiable in the domain of the natural numbers.

PROOF: (An immediate corollary of the completeness proof.)
(We use axiom of choice.)

Since $EzKz$ let a be some object such that Ka. Let f be the function that gives the u for each x. Take the domain $\{a, f(a), ff(a)),\ldots\}$ closed with respect to f. Now identify this (by remaining) with $\{0,1,2,\ldots\}$. [In the model we do not necessarily have $x' \neq 0$ and $x' = y' \supset x = y$. Thus do not exclude finite models.]

Now consider the conjuction of the conditions above

$$K0$$
$$(\forall x)(\forall y)Mxx'y$$
$$(\forall x)(\forall y)Nxy$$

Thus the condition for the domino set is of the form

(1) $$K0 \wedge \forall x \forall y Mxx'y$$

But this is satisfiable iff the domino set has a solution.

Hence we <u>cannot</u> determine whether or not it is satisfiable.

Hence we cannot determine whether or not

$$\forall z K'z \vee \exists x \forall u \exists y Mxuy$$

is a <u>theorem.</u>

THE DECISION PROBLEM

The decision problem for the first-order predicate calculus.

To find an effective method to determine for an arbitrary formula of the first-order predicate calculus, whether or not it is a theorem. (Or, equivalently, whether or not it is satisfiable.)

The classic problem, also known as the ENTSCHEIDUNGS-PROBLEM was first shown to be unsolvable in 1936 by Church and Turing. The two proofs were quite different. Church's proof uses the undecidability of elementary number theory (Gödel's result). For if we take any undecidable statement, prefix the conjunction of the axioms for number theory, and remove function symbols and constants we obtain an expression of the first-order predicate calculus. If it were decidable, then number theory would also be decidable, contradicting the Gödel result. (In order to make this proof go though, we require a finite axiomatization of number theory. Robinson's system, given in Mendelson, is an example of such a finite axiomatization.)

Turing's proof is independent of Gödel's result and uses the halting problem for Turing machines (which in fact were invented for this purpose). The proof given by Turing works directly with the Turing machines, without dominoes, and gives a weak prefix. A much less complicated proof by Büchi along the same lines, gives the $E \wedge AEA$ (satisfiability) result. The method of dominoes was used in the Kahr-Moore-Wang proof for AEA (satisfiability).

The unsolvability of the decision problem for the first-order predicate calculus follows *a fortiori*.

OUTLINE OF THE KAHR-MOORE-WANG (-BERGER) PROOF
OF UNSOLVABILITY OF EAE .

1. The halting problem for TM with blank tape is unsolvable.

2. The complete configurations of any TM can be represented by squares in the plane.

3. The graphic representation of the TM can be described by a domino set. The conditions on the solvability of the dominoes can be expressed in terms of the predicate calculus. (For the original proof, a diagonal-constrained solution is used ... there are an infinite number of the copies of the TM at any one time. For the unrestricted solution there are also an infinite number, but their placement on the plane is different. We demonstrated above the method of the proof in a simpler case, using the origin-constrained problem, a single representation of the TM and settled only for the $A \wedge EAE$ case.)

4. An expression AEA B (B q-free) can be written describing the domino set. Such that AEA B is satisfiable iff the quadrant can be filled.

5. But AEA B is not satisfiable iff the TM halts. Hence EAE -B is provable iff TM halts (by the Gödel completeness theorem).

6. Therefore if EAE were decidable, we could decide the halting problem.

RECALL: DEFINITION OF A REDUCTION CLASS.

A class C of formulas of the first-order predicate calculus is a <u>reduction class</u>, if for every formula F we can find a formula F' in C, such that F is a theorem if and only if F' is a theorem.

Example: The class of formulas in Skolem Normal Form is a reduction class. $(Ex_1 \ldots Ex_m Ay_1 \ldots Ay_n M$. This was proven before.)

Note that these methods show that the class EVE is a <u>reduction class</u>.

1. Construct the TM which carries out the Herbrand Expansion for the given formula F .
2. This TM will halt if and only if the given formula F is a theorem.
3. Use the above process to construct a formula F for that TM. The formula is satisfiable if and only if the TM does not halt. Hence, its negative F' is a theorem if and only if the given formula F is a theorem.

Note also that the <u>dyadic</u> predicate calculus has been shown to be a reduction class.

But there are solvable subcases.

SOLVABLE AND UNSOLVABLE CASES

At this point we have determined all of the prefix-defined cases of the decision problem: (for provability)

SOLVABLE	UNSOLVABLE
$Ax_1 \cdots Ax_m Ey_1 \cdots Ey_n$	AzKz ExAuEyMxuy which gives
$Ax_1 \cdots Ax_m EyAz_1 \cdots Az_n$	AzExAuEy
$Ax_1 \cdots Ax_m Ey_1 Ey_2 Az_1 \cdots Az_n$	ExAuAzEy
	ExAuEyAz
	$Ex_1 \cdots Ex_m Ay_1 \cdots Ay_n$ (S.N.F.)
	ExAuEy

Thus, we have settled all prefix cases. For, (1) adding a quantifier can never make a case solvable, and (2) <u>ExEuEyAz</u> follows as the S.N.F. of ExAuEy .