Propositional and first-order predicate calculus applied to the study of program correctness.

Kenneth Gene Krauss

Follow this and additional works at: http://preserve.lehigh.edu/etd
Part of the Computer Sciences Commons

Recommended Citation
PROPOSITIONAL AND FIRST-ORDER PREDICATE
CALCULUS APPLIED TO THE STUDY OF PROGRAM CORRECTNESS

by

Kenneth Gene Krauss

A Thesis
Presented to the Graduate Committee
of Lehigh University
in Candidacy for the Degree of
Master of Science
in
Computing Science

Department of Mathematics
Division of Computing and Information Science
Lehigh University
1982
This thesis is accepted and approved in partial fulfilment of the requirements for the degree of Master of Science.

May 13, 1982

Professor in Charge

Head of the Division
Acknowledgment

The author wishes to thank Professor Samuel L. Gulden for his excellent instruction which has made this paper possible.
# TABLE OF CONTENTS

1. Abstract ........................................... 1
2. Introduction ....................................... 2
3. Background ......................................... 3
4. Propositional Calculus ............................. 9
5. Predicate Calculus ................................ 41
6. Flowchart Programs ............................... 85
7. Conclusions ........................................ 108
8. List of References ................................. 110
9. Vita ................................................. 111
1. ABSTRACT

The approach that should be taken in order to be able to formally prove the correctness of a computer program is at present an open question. This paper studies the use of propositional and first-order predicate calculus for establishing program correctness. Some background in these calculi is presented first and then their application to program correctness considered. It becomes apparent that the present methods of writing programs and the structures of programming languages must be altered for a complete application of the formal logic to program correctness. For this reason flowchart programming is described and the methods established are applied to flowchart programming. Programs are seen as determined by their input predicates and output predicates. The halting problem exists and can be ignored if partial correctness is sufficient but must be considered for total correctness.
2. INTRODUCTION

At present there is no way to formally prove the correctness of a computer program. At least not for a program of any sophistication. Many programs are in use for long periods of time before situations arise which prove them incorrect. This, of course, implies that satisfactory processing over a period of time does not prove a program is correct.

To approach this topic we must formally define what is meant by program correctness. In addition to this, a formal language suited to the study must be adopted. This paper addresses itself to the use of propositional and predicate calculi as the formal structures through which program correctness may be approached.

In order to do this, we must first establish those results of the calculi which are relevant. A considerable amount of notation must also be introduced. Having these things out of the way, the results and techniques are applied to a rather formal style of programming referred to as flowchart programming.
3. BACKGROUND

In this section we present some of the notation and terminology used in this paper. Because the notation that is used in this field is not yet standard, additional notation and terminology will need to be explained as it is developed and used throughout the paper.

We begin with the following definitions.

For each integer \( N \geq 0 \)

Let \([n] = \{1, 2, \ldots, n\}\) \( [0] = \emptyset \)

\([n] = \{i | i \text{ integer}, 0 < i \leq n\}\)

DEF. \( X^n \) to be the set of all functions \( a : [n] \rightarrow X \)

Let \( \lambda : [0] \rightarrow X \)

\( X^0 = \{\lambda\} \) \( \emptyset^0 = \emptyset \) for \( n > 0 \) \( \emptyset^0 = \{\lambda\} \)

Example: \( X = \{a, b, c\} \) \( n = 2 \) then \( X^n \) is the set of all functions \( a : [2] \rightarrow X \). These could be listed as:

\( \text{listed as: } aa \ ab \ ac \)
\( \quad \text{bb } ba \ bc \quad \text{i.e. all pairs of} \)
\( \quad \text{cc } ca \ cb \quad \text{things in } X. \)

We also write \( a = a(1) \ a(2) \ldots a(n) \) \( n > 0 \)

\( a = a_1 \ a_2 \ldots a_n \) \( n > 0 \) \( a(i) = a_i \)

\( a = (a_1, a_2, a_3 \ldots a_n) \)

When we allow these notations we also identify \( X^1 \) with \( X \).
A formal language on \( X \) is a subset of \( X^* \).

If \( a \in X^m \), \( b \in X^n \), define \( a \cdot b \in X^{m+n} \) by

\[
a \cdot b (i) = \begin{cases} a(i) & 1 \leq i < m \\ b(i-m) & m+1 \leq i \leq m+n \end{cases}
\]

Example: \( X = \{a, b, c\} \)

\[
\begin{align*}
\text{a} & \in X^2 \\
\text{b} & \in X^3 \\
\text{a} & = \text{ab} \\
\text{b} & = \text{bca}
\end{align*}
\]

\[a \cdot b \in X^{2+3} = X^5\]

\[
\begin{align*}
a \cdot b(1) &= a(1) = a \\
a \cdot b(2) &= a(2) = b \\
a \cdot b(3) &= b(1) = b \\
a \cdot b(4) &= b(2) = c \\
a \cdot b(5) &= b(3) = a
\end{align*}
\]

Lemma: If \( a, b, c \in X^* \) then

(i) \( a \cdot (b \cdot c) = (a \cdot b) \cdot c \)

(ii) \( a \cdot \lambda = a = \lambda \cdot a \)

i.e. \( X^* \) is the free monoid on \( X \).

PROOF: (i) Let \( a \in X^m \), \( b \in X^n \), \( c \in X^p \)

Then \( a \cdot (b \cdot c) \) (i) = \[
\begin{align*}
\{ & a(i) & 1 \leq i < m \\
& b \cdot c(i-n) & m+1 \leq i \leq m+(n+p) \}
\end{align*}
\]

\[
(a \cdot b) \cdot c \) (i) = \[
\begin{align*}
\{ & a \cdot b(i) & 1 \leq i < m+n \\
& c(i-m-n) & m+n+1 \leq i \leq (m+n) + p \}
\end{align*}
\]
If $1 \leq i \leq m$ then
\[ a \cdot (b \cdot c) (i) = a(i) \]
\[ (a \cdot b) \cdot c (i) = a \cdot b(i) = a(i) \]

If $m + 1 \leq i \leq m + n$ then
\[ a \cdot (b \cdot c) (i) = b \cdot c(i - m) = b(i - m) \text{ since } 1 \leq i - m \leq n \]
\[ (a \cdot b) \cdot c (i) = a \cdot b(i) = b(i - m) \]

If $m + n + 1 \leq i \leq m + n + p$ then
\[ a \cdot (b \cdot c) (i) = b \cdot c(i - m) = c(i - m - n) \text{ since } n + 1 \leq i - m \leq n + p \]
\[ (a \cdot b) \cdot c (i) = c(i - m - n) \]

(ii) Note $\lambda \in \lambda^0$
\[ a \cdot \lambda(i) = \begin{cases} a(i) & 1 \leq i \leq m \\ \lambda(i - m) & m + 1 \leq i \leq m + 0 \end{cases} \]
\[ \lambda \cdot a(i) = \begin{cases} \lambda(i) & 1 \leq i \leq 0 \\ a(i - 0) & 0 + 1 \leq i \leq 0 + m \end{cases} \]

So $a \cdot \lambda = \lambda \cdot a = a$ for all $i$ such that $1 \leq i \leq m$
and is undefined otherwise.

When no confusion occurs we write $ab$ for $a \cdot b$

If $a \in X^*$, define $a^0 = \lambda$, $a^{n+1} = a \cdot a^n = aa^n$

Example: $a \in X^2$, $a = bc$ then $a^3 = bcbbc$

Let $N$ be the set of non-negative integers. If $X$ is a set, an arity function on $X$ is a function $\rho: X \to N$. 

5
A subset  \( a \subseteq X^* \) is said to have property \( F(\cdot) \) iff for any \( u \in X \) with \( \rho(u) = n \) and \( v_1 \ldots v_n \in a \) it follows that \( uv_1 \ldots v_n \in a \)

Note: \( v_1, v_2, \ldots, v_n \) are elements of \( X \).

Corollary: If \( X_0 = \{ x | \rho(x) = 0 \} \) then \( X_0 \subseteq a \) assuming \( a \) has property \( F(\rho) \).

Note: \( X^* \) has property \( F(\rho) \).

Let \( F(X, \rho) = \bigcap \{ a | a \subseteq X^*, a \text{ has property } F(\rho) \} \)

observe \( F(X, \rho) \) has property \( F(\rho) \).

\( F(X, \rho) \) is the functional \( \rho \)-closure on \( X \).

Lemma: \( F(X, \rho) = \emptyset \) iff \( X_0 = \emptyset \)

This is restated and proven on page 8.

If \( X_0 = \emptyset \) then \( \emptyset \) satisfies \( F(\rho) \). This follows from the fact that if \( b \in \emptyset \) then \( ab \in \emptyset \). In other words, for each \( b \) we have \( b \in \emptyset \# ab \in \emptyset \) since \( b \in \emptyset \) must be false.

Example: Let \( X = \{ a, b \} \) and \( \rho(a) = \rho(b) = 1 \).

Then \( \emptyset \) satisfies \( F(\rho) \) since for each \( c \in \emptyset \) it follows that \( ac \in \emptyset \) and \( bc \in \emptyset \) and so on.

From this we see that if \( X_0 = \emptyset \) then \( F(X, \rho) = \emptyset \) because the empty set, \( \emptyset \), is in the intersection with the other sets that satisfy \( F(\rho) \).
Define \( a_0 = X_0 \) and suppose we have defined \( a_0, \ldots, a_k \) for \( k \geq 0 \) then let \( a_{k+1} \) be the set of all elements of \( X^* \) of the form \( uv_1 \ldots v_n \) where \( \rho(u) = n \geq 0, \)
\( v_1, \ldots, v_n \in a_0u \).

**Example:** \( X = \{a, b, c, 0, 1, 2, +, *, =\} \)

\[\begin{align*}
\rho(a) &= \rho(b) = \rho(c) = \rho(0) = \rho(1) = \rho(2) = 0 \\
\rho(+) &= \rho(*) = \rho(=) = 2
\end{align*}\]

Some allowable strings are: \(+ab = a0\)
\(++1bc\)
\(+1b*0c\)

notice \(++1bc = (1 + b) + c\)
\(+1b*0c = (1 + b) + (0*0)\)

We are defining prefix notation. Notice we don't need the parenthesis. We will assume that all things can be written this way. All of our function notation is prefix.

This gives us a formal way to write things.

i.e. languages.

Let \( a^* = \bigcup \{a_k | k \geq 0\} \)

**Lemma:** \( F(X, \rho) = \bigcup \{a_k | k \geq 0\} = a^* \)
PROOF: \(a^*\) has property \(F(\rho)\).

Let \(\rho(u) = n\) and \(v_1, \ldots, v_n \in a^*\).

Then for each \(i\) chose \(k_i\) such that \(v_i \in a_{k_i}\).

Let \(k = \max (k_1, \ldots, k_m)\) then \(uv_1 \ldots v_n \in a_{k+1}\)
and hence \(uv_1 \ldots v_n \in a^*\). Therefore:

\[F(X, \rho) \subseteq a^*.\]

Now observe that \(a_0 \subseteq F(X, \rho)\). Suppose we
have shown that \(a_0, \ldots, a_k \subseteq F(X, \rho)\) for some
\(k \geq 0\). Then by definition of \(a_{k+1}\) and the fact
that \(F(X, \rho)\) has property \(F(\rho)\) it follows that
\(a_{k+1} \subseteq F(X, \rho)\). Hence, \(a^* \subseteq F(X, \rho)\).

Therefore, \(F(X, \rho) = a^*\) by double inclusion.

Corollary: \(F(X, \rho) = \emptyset\) if \(X_0 = \emptyset\)

**PROOF:** \(a^* = \emptyset\) if \(X_0 = \emptyset\)

**Note:** If \(a_0 = \emptyset\) then \(a_1 = \emptyset\) and if \(a_0 = a_1 = \emptyset\)
then \(a_2 = \emptyset\) etc. Recall \(a_0 = X_0\) so if
\(X_0 = \emptyset\) then \(a^* = \emptyset\)

Also notice that \(A_1 \subseteq A_2 \subseteq A_3 \subseteq \) etc.
4. PROPOSITIONAL CALCULUS

Let \( V \) be a non-empty set called variables and \( \rightarrow \) and \( \pm \) are not elements of \( V \).

Let \( X = V \cup \{ \rightarrow, \pm \} \).

Let \( \rho : X \rightarrow \mathbb{N} \) be defined by \( \rho(A) = 0 \) for \( A \in V \) and \( \rho(\rightarrow) = 1 \) and \( (\pm) = 2 \).

Let \( WF(V) = F(X, \rho) \). (This gives us the well formed formulas)

Example: \( \pm A \rightarrow B \) this is \( A \pm \rightarrow B \)
\( \pm A \pm BC \) this is \( A \pm (B \pm C) \)

We assume the cardinality of \( V = \infty \).

We use the letters \( A, B, C \) to stand for meta-linguistic variables ranging over \( WF \).

We introduce the informal notations \( A \pm B \) for \( \pm AB \) where the latter names the result of obvious concatenation.

We also introduce parenthesis, brackets, braces, etc. where necessary.

Example: \( A \pm B \pm C \) could mean \( \pm A \pm BC \) or \( \pm \pm ABC \) depending on where we put the parenthesis.

We must have a way of assigning meaning.

A mapping into a set gives the set as meaning.
A valuation on WF is a function \( H:WF \rightarrow \{0, 1\} \) with the following properties.

\[
H(\rightarrow A) = 1 - H(A)
\]

\[
H(\pm AB) = \max (H(\rightarrow A), H(B))
\]

This is equivalent to the Truth Table.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>A \pm B</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

Where if A is true then \( H(A) = 1 \).

Suppose you have \( X, \rho \)

\[
X_0 = \{x | \rho(x) = 0\} \quad \text{and} \quad F:X_0 \rightarrow Y
\]

If \( \rho(u) = n > 0 \) we are given a function \( f_u:Y^n \rightarrow Y \).

Can we extend \( F \) to \( F \) where \( F(uv_1 \ldots v_n) = f_u(F(v_1) \ldots F(v_n)) \)? This would mean that at the bottom of the tree of formulas are only things of arity zero. In other words at the leaves.
Define $\mathcal{J} \subseteq \text{WF}$ by $\mathcal{J} = \{ A | A \in \text{WF}, H(A) = 1 \}$ for all valuations $H$.

**Example:** $H(A \neq A)$ or $H(A \neq AA) = 1$ whether $H(A) = 1$ or $H(A) = 0$. This is equivalent to the Truth Table.

<table>
<thead>
<tr>
<th>A</th>
<th>$A \neq A$</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

If one means true and zero false, then only those things that evaluate to one are true. Only here do we know what true things are. When the calculus is a little complicated we have non-provable but true things. This is the last calculus in which the meaning is simple or mechanical. In the real world meaning changes.

**Example:** "Liberty smokes procrastination."

This could be true if Liberty is a person and Procrastination is a cigarette.

We would like to have a calculus to use to determine if a program is correct. In other words we would like to be able to write program verification programs. This can only be done with partial success. For example it is known that there is no solution to the "stopping" or "halting" problem.
Tautologies - each of the following names an infinite set of tautologies or well formed formulas.

(i) \( A \neq A \)
(ii) \( A \neq (B \neq A) \)
(iii) \( [A \neq (B \neq C)] \neq [(A \neq B) \neq (A \neq C)] \)
(iv) \( [\neg B \neq \neg A] \neq [A \neq B] \)
(v) \( \neg B \neq (B \neq \lambda) \)
(vi) \( (B \neq \neg A) \neq [(B \neq A) \neq \neg B] \)

A quick way to establish these is to assume that the expression is false and lead to a contradiction.

**Example:** \( [A \neq (B \neq C)] \neq [(A \neq B) \neq (A \neq C)] \)

If this valuates to 0 then the left must valuate to one and the right side must valuate to 0. For the right side to valuate to 0, \( (A \neq C) \), must valuate to 0 which means \( H(A) = 1 \) and \( H(C) = 0 \). Also, \( (A \neq B) \), must valuate to 1 so \( H(B) = 1 \). Now, we see that the left side can not valuate to 1 since \( H(A) = 1 \), \( H(B) = 1 \) and \( H(C) = 0 \).

The semantics of the propositional calculus is determinable mechanically. There is no internal analysis of the statements. We need to study the axiomatic method or method of derivations.
Derivations from Axioms:

Axioms - each represents an infinite set of well-formed formulas.

A_1 \quad A \vdash (B \vdash A)
A_2 \quad [A \vdash (B \vdash C)] \vdash [(A \vdash B) \vdash (A \vdash C)]
A_3 \quad [\neg B \vdash \neg A] \vdash [A \vdash B]

Now we need the notion of derivation.

Let \( \Gamma \subseteq WF(v) \) be any subset of well-formed formulas on \( v \). If \( A \in WF(v) \) we define \( \Gamma \vdash A \) (read \( \Gamma \) derives \( A \)) to mean there exists a sequence \( D_1, D_2, \ldots, D_m \) of \( WF \) formulas \( (D_i \in WF(v)) \) such that:

1. \( D_m = A \) (\( D_m \) and \( A \) name the same formula)
2. For each \( j, 1 \leq j \leq m \) at least one of the following holds:
   
   (a) \( D_j \) is an axiom
   (b) \( D_j \) is in \( \Gamma \)
   (c) There exists \( p, q < j, p \neq q \), such that \( D_q = D_p \vdash D_j \) This we call the rule of detachment.

We now wish to establish a set of derived formulas.

Let \( \Gamma = \{ A, A \vdash B \} \) then \( \Gamma \vdash B \)

PROOF: \( D_1 = A \) by 2(b)
\( D_2 = A \vdash B \) by 2(b)
\( D_3 = B \) by 2(c)

If \( \Gamma \vdash B \) and \( \Gamma = \emptyset \) then we write \( \vdash B \) i.e. \( B \) is derivable from axioms.
Axioms are derivable: \( \vdash A_1, \vdash A_2, \vdash A_3 \) by 2(a).

\( T_1: \vdash A \leq A \)

**PROOF:**
1. \( A \leq [(A \leq A) \leq A] \)
   by \( A_1 \)
2. \( [A \leq ((A \leq A) \leq A)] \leq \\
   [(A \leq (A \leq A)) \leq (A \leq A)] \)
   by \( A_2 \)
3. \( [A \leq (A \leq A)] \leq [A \leq A] \)
   1, 2, detach.
4. \( A \leq (A \leq A) \)
   by \( A_1 \)
5. \( A \leq A \)
   3, 4, detach.

\( T_2: \vdash (B \leq C) \leq [(A \leq B) \leq (A \leq C)] \)

1. \( [A \leq (B \leq C)] \leq \\
   [(A \leq B) \leq (A \leq C)] \)
   by \( A_2 \)
2. \( (1) \leq ((B \leq C) \leq (1)) \)
   by \( A_1 \), where \( (1) \) is the formula of step 1.
3. \( (B \leq C) \leq (1) \)
   1, 2, detach.
4. \( (B \leq C) \leq (A \leq (B \leq C)) \)
   by \( A_1 \)
5. \( (3) \leq ((4) \leq T_2) \)
   by \( A_2 \)
   i.e. Take \( A_2 \) and for:
   A write \( B \leq C \)
   B write \( A \leq (B \leq C) \)
   C write \( (A \leq B) \leq (A \leq C) \)
6. \( (4) \leq T_2 \)
   3, 5, detach.
7. \( T_2 \)
   4, 6, detach.

Things would be a little easier if we had a deduction theorem.
Deduction Theorem

Let \( \Gamma = \{A_1, \ldots, A_n\} \) and suppose \( \Gamma, B \vdash C \) where \( B, C \in WF(v) \) and \( \Gamma, B \) means \( \Gamma \cup \{B\} \) then \( \Gamma \vdash B \neq C \).

Notice how this can be used. We can prove \( T_2 \) by showing that \( B \neq C \), \( A \neq B \), \( A \vdash C \).

1. \( A \)  
   2(b) i.e. \( A \) in is \( \Gamma \)

2. \( A \neq B \)  
   2(b)

3. \( B \)  
   1, 2, detachment

4. \( B \neq C \)  
   2(b)

5. \( C \)  
   3, 4, detachment

6. \( B \neq C, A \neq B \vdash A \neq C \)  
   deduction theorem (DT)

7. \( B \neq C \vdash (A \neq B) \neq (A \neq C) \)  
   DT

8. \( (B \neq C) \neq [(A \neq B) \neq (A \neq C)] \)  
   DT

As another example, observe \( A \vdash B \neq A \)

1. \( A \)  
   2(b)

2. \( A \neq (B \neq A) \)  
   \( A_1 \)

3. \( B \neq A \)  
   1, 2, detachment

Note: If \( A \) is an axiom then \( \vdash B \neq A \)
Proof of the Deduction Theorem

Let $D_1, \ldots, D_m$ be a derivation of $\Gamma, B, \vdash C$.
We show by induction that $\Gamma \vdash B \neq D_j$ for $j = 1, \ldots, m$.
Since $D_m = C$ this will establish that $\Gamma \vdash B \neq C$.

We first consider the case $D_1$. By the definition of a derivation there are 3 possibilities.

(a) $D_1 \in \Gamma$
(b) $D_1$ is an axiom
(c) $D_1 = B$

We consider (a) and (b) together:
1. $D_1 \vdash (B \neq D_1)$  \hspace{1cm} A_1
2. $D_1$ by (a) or (b)
3. $B \neq D_1$  \hspace{1cm} 1, 2, detachment

Hence, $\Gamma \vdash B \neq D_1$. In case (c) we have already shown that $\vdash D_1 \neq D_1$ so $\vdash B \neq D_1$ and hence $\Gamma \vdash B \neq D_1$.

Suppose we have demonstrated that $\Gamma \vdash B \neq D_j$ for all $j$ such that $1 \leq j < k$ and some $k > 1$. In case $D_k$ satisfies (a), (b), or (c) above, then the argument to show $\Gamma \vdash B \neq D_k$ is the same. The only other possibility is that there exists $q, q < k$ such that $D_q = D_p \neq D_k$. By the induction hypothesis we know that:

$\Gamma \vdash B \neq D_p$ and $\Gamma \vdash B \neq D_q$ or in other symbols $\Gamma \vdash B \neq (D_p \neq D_k)$ now $[B \neq (D_p \neq D_k)] \neq$

$[(B \neq D_p) \neq (B \neq D_k)]$ by A_2

$(B \neq D_p) \neq (B \neq D_k)$ \hspace{1cm} detachment
B \models D_k \\
\text{detachment}

hence \Gamma \vdash B \models D_k.

As we have seen, this deduction theorem provides some conveniences for us.

Note the following matters about deductions:

1) If \Gamma, A \vdash B and \Gamma, B \vdash C then \Gamma, A \vdash C

2) If \Gamma, A, B \vdash C then \Gamma, B, A \vdash C

\Gamma, A, B \vdash C gives \Gamma \models A \models (B \models C) from which we can derive C given A and B.

\Gamma, B, A \vdash C gives \Gamma \models B \models (A \models C) from which we can derive C given A and B.

We now need some definitions - meta-linguistic definitions. Recall that A and B range over well formed formulas.

\begin{align*}
VAB & := \# \rightarrow AB & \equiv & \rightarrow A \equiv B \\
\land AB & := \rightarrow \# A \rightarrow B & \equiv & \rightarrow (A \models \rightarrow B) \\
t & := \# AA & \equiv & A \models A \\
f & := \rightarrow \# AA & \equiv & \rightarrow (A \models A) \\
\# AB & := \rightarrow \# \models AB \rightarrow \# BA & \equiv & \rightarrow ((A \models B) \models \rightarrow (B \models A))
\end{align*}

We will use the informal infix notations

A V B read A or B

A \land B read A and B

A \# B read A is equivalent to B

This means that we can substitute the new definitions for the old expressions.
We want to prove the completeness of the propositional calculus. We would like to establish that it is not possible to derive a statement and its negation.

Completeness: Two ways to state it.

1. There does not exist an $A \in WF(v)$ such that $\neg A$ and $\neg \neg A$.

2. For any $A \in WF(v)$, then $\neg A$ iff $A$ is a Tautology.

We first need to derive some more formulas:

Lemma. If $A \neq B$, $B \neq C$, then $A \dashv C$.

Corollary. $(A \neq B) \vdash [(B \neq C) \neq (A \neq C)]$ by D.T.

$(B \neq C) \vdash [(A \neq B) \neq (A \neq C)]$ by D.T.

$(A \neq B) \vdash [A \neq ((B \neq C) \neq C)]$ by D.T.

We could go on permuting these on the left side of the $\neg$ and then using the deduction theorem several times to get six such forms. The important thing is to recognize these forms:

$T_3$:  

1. $(\neg B \neq \neg A) \neq (A \neq B)$ by $A_3$

2. $\neg A \neq (\neg B \neq \neg A)$ by $A_1$

3. $\neg A \neq (A \neq B)$ by D.T.
T₄:  \[ \rightarrow \neg \neg A \neq A \]
1. \[ \rightarrow \neg \neg A \neq (\rightarrow \neg \neg A \neq \rightarrow \neg \neg A) \] by T₃
2. \[ (\rightarrow \neg \neg A \neq \rightarrow \neg \neg A) \neq (\rightarrow \rightarrow \neg \neg A \neq A) \] by A₃
3. \[ \rightarrow \neg \neg A \neq (\rightarrow \rightarrow \neg \neg A \neq A) \]
4. \[ \rightarrow \neg \neg A \neq \rightarrow \neg \neg A \]
5. \[ (3) \neq ((4) \neq T₄) \]
   i.e. take \( A₂ \) and for:
   A write \( \rightarrow \neg \neg A \)
   B write \( \rightarrow \rightarrow \neg \neg A \)
   C write \( \rightarrow \neg \neg A \)
6. \[ T₄ \] by detachment twice

T₅:  \[ \rightarrow \neg \neg A \neq \rightarrow \neg \neg A \]
1. \[ \rightarrow \neg \neg A \neq \rightarrow \neg \neg A \] by T₄
2. \[ (\rightarrow \neg \neg A \neq \rightarrow \neg \neg A) \neq (A \neq \rightarrow \neg \neg A) \] by A₃
3. \[ A \neq \rightarrow \neg \neg A \]

Here are some obvious results at this stage.

\[ \rightarrow \neg \neg (\neg \neg A \neq A) \] since \( t := A \neq A \)
\[ \rightarrow \neg \neg (\neg \neg A \neq A) \] this is by definition
   \[ \neg \neg (\neg \neg A \neq A) \]
1. \[ (\neg \neg A \neq \neg \neg A) \neq \neg \neg (\neg \neg A \neq \neg \neg A) \] by T₅
2. \[ A \neq \neg \neg A \] by T₁
3. \[ \neg \neg (\neg \neg A \neq \neg \neg A) \]
   1, 2, detach.
$T_6 \vdash A \vdash [(A \vdash B) \vdash B]$ is almost trivial. It comes from $A, A \vdash B \vdash B$ and 2 applications of the deduction theorem.

$T_7 \vdash [\vdash \neg A \vdash B] \vdash (\neg \neg B \vdash A)$
1. $\neg \neg B \vdash \neg \neg B$ by $T_5$
2. $\neg \neg B \vdash \neg \neg B, \neg A \vdash \neg A, \neg A \vdash \neg \neg B$ Detach.
3. $(\neg \neg B \vdash \neg \neg B) \vdash [(\neg \neg A \vdash B) \vdash$

$\neg \neg A \vdash \neg \neg B)]$ Deduct.
4. $(\neg \neg A \vdash B) \vdash (\neg \neg A \vdash \neg \neg B)$ 1, 3, Detach.
5. $(\neg \neg A \vdash \neg \neg B) \vdash (\neg \neg B \vdash A)$ by $A_3$
6. $(\neg \neg A \vdash B) \vdash (\neg \neg B \vdash A)$ 4, 5, Deduct.

$T_8 \vdash (A \vdash \neg B) \vdash (\neg B \vdash A)$
1. $\neg \neg A \vdash A$ by $T_4$
2. $\neg \neg A \vdash A, A \vdash \neg B,$

$\neg \neg A \vdash \neg \neg B$ Detach. to D.T. Lemma
3. $(\neg \neg A \vdash A) \vdash [(A \vdash \neg B) \vdash$

$(\neg \neg A \vdash \neg B)]$ Deduct.
4. $(A \vdash \neg B) \vdash (\neg \neg A \vdash \neg \neg B)$ 1, 3, Detach.
5. $(\neg \neg A \vdash \neg B) \vdash (\neg B \vdash A)$ by $A_3$
6. $(A \vdash \neg B) \vdash (\neg B \vdash \neg \neg A)$ 4, 5, Deduct.
T_{8.5} \vdash (A \leftrightarrow B) \leftrightarrow (\rightarrow B \leftrightarrow \rightarrow A)

1. \rightarrow \rightarrow A \leftrightarrow A \\
2. \rightarrow \rightarrow A \leftrightarrow A, A \leftrightarrow B, \rightarrow \rightarrow A \rightarrow B \\
3. (\rightarrow \rightarrow A \leftrightarrow A) \leftrightarrow ((A \leftrightarrow B) \leftrightarrow \\
(\rightarrow \rightarrow A \leftrightarrow B)) \\
4. (A \leftrightarrow B) \leftrightarrow (\rightarrow \rightarrow A \leftrightarrow B) \\
5. (\rightarrow \rightarrow A \leftrightarrow B) \leftrightarrow (\rightarrow B \leftrightarrow \rightarrow A) \\
6. (A \leftrightarrow B) \leftrightarrow (\rightarrow B \leftrightarrow \rightarrow A) \\

T_{9} A, B \vdash \rightarrow \rightarrow (A \leftrightarrow \rightarrow B)

1. A \leftrightarrow [(A \leftrightarrow \rightarrow B) \leftrightarrow B] \\
2. [(A \leftrightarrow \rightarrow B) \leftrightarrow \rightarrow B] \leftrightarrow [B \leftrightarrow \rightarrow (A \leftrightarrow \rightarrow B)] \\
3. A \leftrightarrow [B \leftrightarrow \rightarrow (A \leftrightarrow \rightarrow B)] \\
4. A \\
5. B \leftrightarrow \rightarrow (A \leftrightarrow \rightarrow B) \\
6. B \\
7. \rightarrow \rightarrow (A \leftrightarrow \rightarrow B) \\

Corollary 9.5 A, \rightarrow B \vdash \rightarrow (A \leftrightarrow B)

Note: This is not the same as T_{9} because we do not
have a replacement theorem.

1. A, \rightarrow B \vdash \rightarrow (A \leftrightarrow \rightarrow B) \\
2. B \leftrightarrow \rightarrow \rightarrow B \\
3. A \leftrightarrow B, B \leftrightarrow \rightarrow \rightarrow B, A \rightarrow \rightarrow B \\
4. (B \leftrightarrow \rightarrow \rightarrow B) \leftrightarrow [(A \leftrightarrow B) \leftrightarrow \\
(A \leftrightarrow \rightarrow \rightarrow B)] \\

21
5. \((A \neq B) \neq (A \neq \neg \neg B)\)
6. \([(A \neq B) \neq (A \neq \neg \neg B)] \neq \neg (A \neq B)\)
7. \(\neg (A \neq \neg \neg B) \neq \neg (A \neq B)\)
8. \(\neg (A \neq B)\)

\[T_{10}\]
1. \(\neg B \neq (A \neq \neg \neg B)\)
2. \(\neg (A \neq \neg \neg B) \neq B\)

\[T_{11}\]
1. \([A \neq \neg \neg (B \neq B)] \neq \neg A\)
2. \(A \neq \neg \neg (B \neq B), B \neq B \rightarrow \neg A\)
3. \(A \neq \neg \neg (B \neq B) \leftarrow \neg A\)
4. \([A \neq \neg \neg (B \neq B)] \neq \neg A\)

\[T_{12}\]
1. \(A\)
2. \(B\)
3. \(C\)
4. \(\neg (B \neq \neg \neg C)\)

Given

Detach.

Detach.

by \(T_9\)
Now we arrive at the first important theorem.

Our goal is to prove that if $A$ is a tautology then $\neg A$. This was essentially the first completeness theorem. The proof was first given by Post and Lucasевич independently. We will not do the classical proof.

Theorem: If $\Gamma, \neg A \vdash B$ and $\Gamma, A \vdash B$ then $\Gamma \vdash B$.

PROOF: $\Gamma \vdash \neg A \neq B$ by deduction

$\Gamma \vdash A \neq B$ by deduction

$\vdash (\neg A \neq B) \neq ((A \neq B) \neq B)$ by $T_{13}$

Hence: $\Gamma \vdash B$.

If $A \in WF(v)$ define $W(A)$ as follows: ("weight of formula")

1. If $A \in V$ $W(A) = 0$
2. If $A = \neg B$ $W(A) = W(B) + 1$
3. If $A = \neq CD$ then $W(A) = W(C) + W(D) + 1$

Note: $W$ picks up the total number of $\neg$ and $\neq$ in the formula.
We also need the following definitions.

**DEF:** Let $H: \text{WF}(v) \rightarrow \{0, 1\}$ be a valuation. If $A \in \text{WF}(v)$, define $A_H$ as follows:

- if $H(A) = 1$, $A_H = A$
- if $H(A) = 0$, $A_H = \lnot A$

**Note:** $A$ is a tautology iff $A_H = A$ for all valuations $H$.

**DEF:** If $A \in \text{WF}(v)$ define $\text{Var}(A)$ as follows:

- if $A \in v$, $\text{Var}(A) = \{A\}$
- if $A = \lnot B$, $\text{Var}(A) = \text{Var}(B)$
- if $A = B \lor C$, $\text{Var}(A) = \text{Var}(B) \cup \text{Var}(C)$

**Note:** $\text{Var}$ brings together all of the distinct variables.

**Theorem:** Let $A \in \text{WF}(v)$. Suppose $H$ is a valuation and $\text{Var}(A) = \{X_1, \ldots, X_n\}$, then $X_{1H}, X_{2H}, \ldots, X_{nH} \vdash A_H$.

**PROOF:** We use induction on $W(A)$.

1. If $W(A) = 0$ then $A \in v$ and the theorem reduces to $A_H \vdash A_H$.
2. Suppose for some $m > o$ and all $k$ with $0 \leq k < m$ we have established the result for any $A$ with $W(A) = k$. 

24
Now, assume \( W(A) = m \).

a) \( A = \text{-} B \)

b) \( A = \not\in BC \)

In (a) above we have \( \text{Var}(A) = \text{Var}(B) \) and \( W(B) < W(A) \). Hence, we have by hypothesis that \( X_{1H}, H_{2H}, \ldots, X_{nH} \leftarrow B \). We consider the following two possibilities:

(\( \alpha \)) \( H(A) = 0 \), then \( A_H = \text{-} A \) and \( H(B) = 1 \) since \( H(A) = H(\text{-} B) = 1 - H(B) = 0 \). Also \( A_H = \text{-} \text{-} B \) and \( B_H = B \). By \( T_5 \) we have \( \text{-} B \not\in \text{-} B \) so \( B_H \not\in A_H \). Hence, \( X_{1H}, \ldots, X_{nH} \leftarrow A_H \).

(\( \beta \)) \( H(A) = 1 \), then \( A_H = A \) and \( H(B) = 0 \). Now, \( B_H = \text{-} B = A = A_H \) so \( X_{1H}, \ldots, X_{nH} \leftarrow A_H \).

In (b) above, we consider the following three cases:

(\( \alpha \)) \( H(C) = 1 \). Therefore \( H(A) = 1 \) since \( A = \not\in BC \).

Since \( W(B) \) and \( W(C) < W(A) \) we have by the induction hypothesis that:

\( X_{1H}, X_{2H}, \ldots, X_{nH} \leftarrow B_H \) and

\( X_{1H}, X_{2H}, \ldots, X_{nH} \leftarrow C_H \) since \( \text{Var}(B) \subseteq \text{Var}(A) \) and \( \text{Var}(C) \subseteq \text{Var}(A) \) and \( \text{Var}(A) = \{X_{1H}, \ldots, X_{nH}\} \). In other words we may have more variables in the list than we actually need.
\(C_H > C\) since \(H(C) = 1\) and we know from Axiom 1 that \(\vdash C \neq (B \pm C)\). It follows that \(X_{1H}, \ldots, X_{nH} \vdash B \pm C\) and therefore \(X_{1H}, \ldots, X_{nH} \vdash A\).

Recall \(A = A_H\) since \(H(A) = 1\). Notice that this argument is independent of what \(H\) does to \(B\).

(6) \(H(B) = 0\). Therefore \(H(A) = 1\), \(B_H = \neg B\), and \(A_H = A\). Recall that we have \(\neg \neg B \neq (B \pm C)\) by \(T_3\). Hence, applying the first derivation \(X_{1H}, \ldots, X_{nH} \vdash B_H\), we have \(X_{1H}, \ldots, X_{nH} \vdash B\) so \(X_{1H}, \ldots, X_{nH} \vdash B \pm C\).

Since \(B \pm C = A = A_H\) we are done.

(7) \(H(B) = 1\) and \(H(C) = 0\). Therefore \(B_H = B\), \(C_H = \neg C\), and \(A_H = \neg A\). From the derivations in (6) we have \(\text{Var } (A) \vdash B\) and \(\neg C\). From Corollary 9.5 \(B, \neg C \vdash (B \pm C)\). Since \(\neg (B \pm C) = \neg A = A_H\) we are done.

Example: \(A = X_1 \neq (X_1 \neq X_2)\) where \(H(X_1) = 1\), \(H(X_2) = 0\).

Recall: \(H(\neg A) = 1 - H(A)\) and \(H(\neq AB) = \max (H(\neg A), H(B))\). \(H(X_1 \neq (X_1 \neq X_2)) = \max (H(\neg X_1), H(X_1 \neq X_2)) = \max (1 - H(x_1), \max (H(\neg X_1), H(X_2))) = \max (0, \max (0, 0)) = \max (0, 0) = 0\).
Hence, \( A_H = \neg \lambda = \neg x_1 = (x_1 \neq x_2) \).

The Theorem says:

\[
\begin{align*}
X_1^{H'} & \quad X_2^{H'} \quad A_H & \quad \text{or} \\
\neg x_1 & \quad \neg x_2 \quad \neg (x_1 = (x_1 \neq x_2)) \quad \text{since} \\
X_1^{H'} & = x_1 \quad \text{and} \quad X_2^{H'} = \neg x_2. \quad \text{Here is a derivation.}
\end{align*}
\]

\[
\begin{align*}
\neg x_1 & \quad (\neg x_2 \neq (x_1 \neq x_2))
\end{align*}
\]

**PROOF:**

\[
\begin{align*}
x_1 & \neq (x_1 \neq x_2), \quad x_1 \quad \neg x_2 \\
x_1 & \neq (x_1 \neq x_2) \quad \neg x_1 \neq x_2 \\
\neg [x_1 \neq (x_1 \neq x_2)] & \quad \neg [x_1 \neq x_2] \\
\neg [x_1 \neq x_2] & \quad \neg (x_1 \neq x_2) \\
\neg x_1 & \quad (\neg x_2 \neq (x_1 \neq x_2)) \\
x_1, \quad \neg x_2 & \quad \neg (x_1 \neq x_2) \\
x_1, \quad \neg x_2 & \quad \neg (x_1 \neq x_2)
\end{align*}
\]

We have now completed the ground work to prove the following theorem. The original proof was done by Post around 1912. The proof that is given here was done by Kalmar in the early 1960's.

**Theorem.** If \( A \) is a tautology then \( \neg A \).

**PROOF:** For any valuation \( H \), \( A_H = A \). Let \( H \) be some fixed valuation. If \( \text{Var}(A) = \{ x_1^H, \ldots, x_n^H \} \) we know that \( x_1^H, \ldots, x_n^H \quad \neg A \) from the previous theorem.
Let \( H \) be a valuation defined as follows:

\[
H(X) = H(X) \text{ if } X \in V \text{ and } X \neq X_n,
\]

\[
H(X_n) = 1 - H(X_n),
\]

and extend \( H \) to all of \( WF(V) \).

We also have \( X_1H, \ldots, X_nH \vdash A \) since this is true for any valuation. Now, \( X_jH = X_jH \) for \( j < n \) so \( X_1H, \ldots, X_{n-1}H, X_nH \vdash A \) and \( X_1H, \ldots, X_{n-1}H, X_nH \vdash A \) where \( \{X_nH, X_n\} = \{X_n, \to X_n\} \).

Thus we have \( X_1H, \ldots, X_{n-1}H, X_n \vdash A \)

\[
X_1H, \ldots, X_{n-1}H, \to X_n \vdash A.
\]

We know that if \( r, A \vdash B \) and \( r, \to A \vdash B \) then \( r \vdash B \) by \( T_{13} \). Hence, \( X_1H, \ldots, X_{n-1}H \vdash A \). Since the valuation \( H \) is arbitrary we can continue in this manner and by iteration establish \( \vdash A \). We can also prove the converse of this theorem.

**Theorem.** If \( \vdash A \) then \( A \) is a tautology.

**PROOF:** All of the Axioms are tautologies: If \( A \) is a tautology and \( A \neq B \) is a tautology then \( B \) is a tautology.

From these two theorems we see that a theorem is derivable if and only if it is a tautology.

We now need the following definitions.
DEF. If \( r \subseteq \text{WF}(V) \) we say that \( r \) is consistent iff

There is no \( A \in \text{WF}(V) \) such that \( r \vdash A \) and \( \vdash \neg A \).

Corollary: If \( r \) is not consistent then for any \( B \in \text{WF}(V) \), \( r \vdash B \). This follows from \( \vdash \neg A \vdash (A \neq B) \) which is \( T_3 \).

DEF. Let \( r \subseteq \text{WF}(V) \) and let \( H \) be a valuation.

We call \( H \) a model for \( r \) or we say that \( H \) satisfies \( r \) iff \( H(A) = 1 \) for all \( A \in r \).

In reality one would like to build what looks like a reasonable model for one's theory.

Lemma: If \( H \) is a model for \( r \) and \( r \vdash B \) then \( H(B) = 1 \).

PROOF: Recall \( H(\neg A) = 1 - H(A), H(A \neq AB) = \max(H(\neg A), H(B)) \).

Since \( H \) is a model for \( r \), \( H(A) = 1 \) for all \( A \in r \).

Also, if \( A \) is an axiom then \( H(A) = 1 \) since we can \( \vdash A \) making \( A \) a tautology.

Consider any derivation of \( B \) by \( r \). Let \( D_1, D_2, \ldots, D_m \) be a derivation of \( B \) by \( r \). Recall: \( D_m = B \) and for each \( j, 1 \leq j \leq m \) at least one of the following holds:

a) \( D_j \) is an axiom

b) \( D_j \) is in \( r \)

c) There exist \( p, q < j, p \neq q, \) such that \( D_q = D_p \neq D_j \)
I will show that each $D_j$ is such that $H(D_j) = 1$. If $D_j$ is an axiom then $H(D_j) = 1$. If $D_j \in \Gamma$ then $H(D_j) = 1$. Consider the first formula, $D_j$, in the derivation which is there as a result of $D_q = D_p \pm D_j$ where $p, q < j$ and $D_q$ and $D_p$ are in the derivation. $H(D_p) = H(D_q) = 1$ since $D_p$ and $D_q$ are both either axioms or in $\Gamma$.

Now, $H(D_q) = H(D_p \pm D_j) = \max(H(\to D_p), H(D_j)) = \max(0, H(D_j)) = 1$.

Therefore, $H(D_j) = 1$. It follows that $H(D_j) = 1$ for $1 \leq j \leq m$ since $H(D_k) = 1$ for $1 \leq k < j$.

Hence, $H(D_m) = H(B) = 1$.

If a theory is to be correct there must be a situation in the real world that satisfies it. Given a collection of formulas, does it have a model? The following theorems will help us approach this question.

Theorem:  
\begin{enumerate}
\item[a)] If $H(B) = 1$ then $H(\pm AB) = 1$ for all $A$.
\item[b)] If $H(A) = 0$ then $H(\pm AB) = 1$ for all $A$.
\item[c)] If $H(B) = 1$ then for any $A_1, A_2, \ldots, A_m$
\[ H(\pm A_1 \pm A_2 \pm \ldots \pm A_m B) = 1. \]
i.e. $H([A_1 \pm (A_2 \pm \ldots \pm (A_m \pm B) \ldots)]) = 1$.
\end{enumerate}
d) If $H(A_j) = 0$ for some $j$ where $1 \leq j \leq m$
then $H(\#A_1 \# A_2 \# \ldots \# A_mB) = 1$ for
all $B$.

PROOF: The results of the theorem are considered
obvious.

Theorem. Let $\Gamma = \{A_1, \ldots, A_n\}$ then $\Gamma$ has a model
iff $\Gamma$ is consistent.

PROOF: If $\Gamma$ has a model, then $\Gamma$ is consistent. For
if $\Gamma \vdash A$ and $\Gamma \vdash \neg A$ we must have $H(A) = 1$
and $H(\neg A) = 1$ which is impossible.

Suppose $\Gamma$ has no model. This is, for any
evaluation $H$ there is a $j$, $1 \leq j \leq n$ such that $H(A_j)$
$= 0$. Let $B$ be any formula. Then for any $H$, $H(A_1 \#$
($A_2 \# \ldots \# (A_n \# B) \ldots ) = 1$. Thus $A_1, A_2, \ldots A_n$
$\vdash B$. Hence we can derive every formula. For exam-
ple $B$ and $\neg B$. Therefore $\{A_1, \ldots, A_n\}$ is incon-
sistent. It follows that if $\Gamma$ is consistent then $\Gamma$
has a model.

It is now our intention to prove the theorem given
above for infinite sets. In order to consider this
problem we need to assume the axiom of choice.

Axiom of Choice

If you have an infinite collection of sets then
you can build an infinite set by taking one thing from
each.
Notice that if I have an infinite set of formulas
the argument given in the previous theorem will not
hold because of the infinite implication. We have
never assumed that our total set of formulas was count-
able. We will need the following definitions.

DEF. A partial order, p.o., on a set X is a relation
such that:

1. for some \( x, y \in X \), \( x \leq y \) holds.
2. for all \( x \in X \), \( x \leq x \).
3. for all \( x, y, z \in X \)
   if \( x \leq y \) and \( y \leq z \) then \( x \leq z \)
4. for all \( x, y \in X \) if \( x \leq y \) and \( y \leq x \) then
   \( x = y \).

We are using the symbol, \( \leq \), to represent the re-
lation.

Example: of a partial ordering.

Consider the points on the real plane and define
\( (r_1, r_2) \leq (s_1, s_2) \iff r_1 \leq s_1 \) and \( r_2 \leq s_2 \).

Notice that not all points have the relation, \((1, 0)\)
and \((0, 1)\) for example, but the conditions for a
partial ordering are obviously met.

A partial ordering on \( X \) is called a total order
iff for any \( x, y \in X \) either \( x \leq y \) or \( y \leq x \). We write
\( (X, \leq) \), to mean a set \( X \) with a p.o. on \( X \). We some-
times refer to this partially ordered set as the p.o. set $x$. If $x$ is a p.o. and $A \subseteq x$ then we say that $u \in x$ is an upper bound of $A$ iff $x \leq u$ for all $x \in A$.

**DEF.** If $x$ is a partially ordered set and $A \subseteq x$ then we say that $u \in x$ is an upper bound of $A$ iff $x \leq u$ for all $x \in A$.

**DEF.** An element $x \in x$ is called maximal iff for any $y \in x$ if $x \leq y$ then $x = y$.

We now state an axiom which is equivalent to the Axiom of Choice. However, the following axiom is easier to use in the proofs which we plan to do.

**Zorn's Axiom** - If $x$ is a partial ordering and every totally ordered subset of $x$ has an upper bound, then $x$ has a maximal element.

We will now show that the axiom of choice can be derived from Zorn's Lemma or Zorn's Axiom. First we will restate the axiom of choice in a more complete manner.

**Axiom of Choice.** Let $a$ be a set of sets such that for any $x_1, x_2, x \in a$ if $x_1 \neq x_2$ then $x_1 \cap x_2 = \emptyset$, $a \neq \emptyset$, and for any $x \in a$, $x \neq \emptyset$, then there is a set $Y$ such that $Y \cap x$ consists of a single element for each $x \in a$. 

33
The technique which will be used in the following theorem is a powerful tool and is often useful. Theorem. Zorn's Axiom implies the Axiom of Choice. PROOF: Let \( K \) be the collection of all sets \( \mathcal{Z} \) such that:
1. \( \mathcal{Z} \subseteq \bigcup \alpha \)
2. for each \( x \in \alpha \), \( x \cap \mathcal{Z} \) contains at most one element.
3. \( \mathcal{Z} \neq \emptyset \)

To see that \( K \neq \emptyset \) observe that for each \( x \in \alpha \) and \( x \in x \), \( \{x\} \in \mathcal{K} \). Next, observe that set inclusion, \( \subseteq \), is a partial ordering on \( \mathcal{K} \). Let \( L \subseteq \mathcal{K} \) be a totally ordered subset of \( \mathcal{K} \). \( L \) is a sub-collection of sets in \( \mathcal{K} \). Now, let \( W = \bigcup L \). Observe that \( W \in \mathcal{K} \), for if \( W \notin \mathcal{K} \) then there exists an \( x \in \alpha \) and \( x_1, x_2 \in W \cap x \) such that \( x_1 \neq x_2 \). Thus we could find \( A, B \in L \) such that \( x_1 \in A \) and \( x_2 \in B \). Since \( L \) is totally ordered either \( A \subseteq B \) or \( B \subseteq A \). Hence \( x_1, x_2 \) are both in either \( A \) or \( B \). It follows that either \( A \) or \( B \) intersects with \( x \) at two points, \( x_1 \) and \( x_2 \). This contradicts that \( A \) and \( B \) are in \( \mathcal{K} \) so \( W \in \mathcal{K} \). Notice we needed the argument given because we do not have \( \mathcal{K} \) closed under union, and \( W \) is a union of sets in \( \mathcal{K} \). Since \( W = \bigcup L \) and \( w \in \mathcal{K} \), \( W \) is an upper bound for
1. in \( K \). By Zorn's axiom \( K \) has a maximal element \( Y \).

\( Y \) is the required set to satisfy the axiom of choice, for otherwise there is an \( x \in \alpha \) such that \( Y \cap x = \emptyset \).

For each \( x \in \alpha \), \( Y \cup \{x\} \in K \), but \( Y \subseteq Y \cup \{x\} \) and \( Y \neq Y \cup \{x\} \) which contradicts \( Y \) being the maximal element.

The next theorem implies that consistency is a finite matter. The number of subsets involved may be infinite, however.

**Theorem.** Let \( \Gamma \subseteq WF(v) \) then \( \Gamma \) is consistent iff each finite subset of \( \Gamma \) is consistent.

**Proof:** If \( \Gamma \) is consistent then any finite subset of \( \Gamma \) is consistent. Now, suppose that every finite subset is consistent but we can find formula \( A \) such that \( \Gamma \vdash A \) and \( \Gamma \vdash \neg A \).

Let \( D_1, D_2, \ldots, D_p \) by the derivation of \( \Gamma \vdash A \).

\( E_1, E_2, \ldots, E_q \) be the derivation of \( \Gamma \vdash \neg A \).

Let \( \Gamma_0 \) be the formula of \( \Gamma \) which appear in \( \{D_1, \ldots, D_p, E_1, \ldots, E_q\} \). Then \( \Gamma_0 \vdash A \) and \( \Gamma_0 \vdash \neg A \). This is a contradiction because every finite subset of \( \Gamma \) is supposed to be consistent.
We now wish to relate the notion of consistency with the notion of model. The next theorem does this.

Theorem. If $\Gamma \subseteq \text{WF}(v)$ then $\Gamma$ is consistent iff $\Gamma$ has a model.

**PROOF:** If $\Gamma$ has a model then obviously $\Gamma$ is consistent.

Now, suppose $\Gamma$ is consistent. Let $K$ be the set of all pairs $(H, S)$ where $H$ is a valuation, $S \subseteq V$, possibly $S = \emptyset$, and for each finite subset $\Gamma_0 \subseteq \Gamma$ there is a model $H'$ for $\Gamma_0$ such that on $S$, $H' = H$.

**Note:** $K \neq \emptyset$ since $(H, \emptyset) \in K$ for each valuation $H$.

**DEF.** $(H_1, S_1) \leq (H_2, S_2)$ iff $S_1 \subseteq S_2$ and $H_2$ agrees with $H_1$ on $S_1$. Note that this is a partial ordering on $K$. Let $L \subseteq K$ be a totally ordered subset. Define $(\mathcal{H}, \mathcal{S})$ as follows:

$$\mathcal{S} = \bigcup\{ S \mid (H, S) \in L \}$$

If $x \in \mathcal{S}$ then we can find $(H, S) \in L$ such that $x \in S$. Define $\overline{H}(x) = H(x)$. Let $\overline{H}(x) = 0$ for $x \notin \mathcal{S}$.

Let $\Gamma_1 \subseteq \Gamma$ be a finite subset of $\Gamma$.

$$\Gamma_1 = \{ A_1, A_2, \ldots, A_m \} \text{ where } A_j \in \Gamma \text{ for } 1 \leq j \leq m.$$  

Let $V_o = \text{Var}(A_1) \cup \text{Var}(A_2) \cup \ldots \cup \text{Var}(A_m)$ $V_o$ is finite.
Let \( V_1 = V_0 \cap S \), then there is a \((H_3, S_3) \in L\) such that \( V_1 \subseteq S_3 \). \( r_1 \) has a model \( H' \) which agrees with \( H_3 \) on \( S_3 \). Hence, \( H' \) agrees with \( H \) on \( S \). Therefore, \((H, S)\) is an upper bound for \( L \).

Note: All the applications of Zorn's Axiom are now available.

Thus \( K \) has a maximal element \((H^*, S^*)\). We will show that \( S^* = V \).

\( S^* = V \) for if not, let \( x \in V \sim S^* \). Then there exists a finite subset \( r_0 \subseteq r \) such that no model of \( r_0 \) agrees with \( H^* \) on \( S^* \cup \{x\} \). Thus if \( H' \) is a model of \( r_0 \) which agrees with \( H^* \) on \( S^* \) then \( H'(x) = 1 - H^*(x) \). Note: We know \( H' \) exists. Let \( r_1 \subseteq \Gamma \) be any finite subset, then \( r_0 \cup r_1 \) has a model \( H'' \) which agrees with \( H^* \) on \( S^* \) and hence \( H''(x) = 1 - H^*(x) \).

It follows that \( H'' \) is a model of \( r_1 \) which agrees with \( H^* \) on \( S^* \) and \( H''(x) = 1 - H^*(x) \). Note: \( r_1 \) is arbitrary.

Let \( \tilde{H} \) be defined by \( \tilde{H}(y) = H^*(y) \) for \( y \neq x \) and \( \tilde{H}(x) = 1 - H^*(x) \). Then every finite set \( r_1 \) has a model which agrees with \( \tilde{H} \) on \( S^* \cup \{x\} \). But, \((H^*, S^*) \preceq (\tilde{H}, S^* \cup \{x\})\) and they are unequal. This contradicts \((H^*, S^*)\) being maximal.
Hence, $S^* = V$ and if $A \in \Gamma$ then $H^*(A) = H'(A) = 1$ since $\{A\}$ is a finite subset of $\Gamma$ and $H^*$ agrees with $H'$ on $V$.

**DEF.** Let $\leq$ be a partial ordering on $x$ and define by $x < y \iff (x \leq y) \land (x \neq y)$. We call $\leq$ a strong partial ordering with the properties:

1) $x < y \neq (x = y) \land (y < x)$
2) $(x < y) \land (y < z) \neq (x < z)$

**Theorem.** Let $\leq$ be a strong partial ordering on $x$. Then there is a strong total order $\leq'$ on $x$ such that $x < y \neq x \leq' y$. (This is not too difficult for finite sets and not necessarily unique.) We see that for finite sets we could reflect onto a line to obtain the indicated topological ordering. This preserves the partial ordering and gives a total order. See the diagram below.

Notice that the partial order is preserved in the total order.

The formal proof of the theorem follows:
PROOF: For the finite case:

Let \((x, <)\) be a finite partially ordered set. We may assume \(x = \{a_1, a_2, \ldots, a_n\}\). If \(x \in x\) then define the height of \(x\) to be the number of elements in the largest sequence \(x_1, x_2, \ldots, x_k\) such that \(x_1 < x_2 < \ldots < x_k = x\). Let \(h(x)\) be the height of \(x\). Note: if \(x < y\) then \(h(x) < h(y)\). Let \(r(a_j) = h(a_j) + 1/2\). In this way each \(a_j\) is assigned a distinct rational number and hence we have a total order. As described in the previous diagram, we have jiggled each one a little bit to get the total order.

Now we will do the proof in general.

PROOF: Let \(V = x x x = ((a, b) | a, b, c \in x)\)

Consider the \(WF(v)\). Form \(\Gamma\) as follows:

1. \(\Gamma\) must contain all the formulas of the form \(\rightarrow (a, a)\).
2. \(\Gamma\) must contain all the formulas \((a, b)\) where \(a < b\).
3. \(\Gamma\) must contain all the formulas of the form \((a, b) \neq (b, a)\).
4. \(\Gamma\) must contain all the formulas of the form \((a, b) \land (b, c) \neq (a, c)\).
5. \(\Gamma\) must contain all the formulas of the form \((a, b) \lor (b, a)\).
Let \( \Gamma_0 \subseteq \Gamma \) be a finite subset and let \( a_1, \ldots, a_n \) be the distinct points of \( x \) which appear in the formulas of \( \Gamma_0 \). Extend \( \Gamma_0 \) to \( \Gamma_1 \) if necessary so that all the formulas of the type 1 to 5, listed above, on \( a_1, \ldots, a_n \) are included. Any model satisfying \( \Gamma_1 \) defines a total order on \( a_1, \ldots, a_n \). By an informal argument such a model exists. Hence \( \Gamma \) is consistent since every finite subset is consistent. Hence \( \Gamma \) has a model which then defines a total order on \( x \).
5. PREDICATE CALCULUS

We are now finished with our view of propositional calculus as a separate entity. It shall now be considered as a part of a larger scheme. We want to be able to say things like, "for each integer \( n \), \( n \) is even.", or "there exists an integer \( n \) which is even". These statements have content and truth value. The variables now range over sets of objects not necessarily statements. We have open, statements, such as

"\( x \) is less than \( y \)" - binary relation \( L(x,y) \)

"\( x \) is even" - unary relation \( E(x) \).

We also now have quantifiers.

We need a notation for these concepts. The classical notation from Principia Mathematica by Russell and Whitehead is:

\((x)\) - for every \( x \)

\((\exists y)\) - there exists a \( y \)

The notation from the Tarski school is:

\((\forall x)\) - for every \( x \)

\((\exists y)\) - there exists a \( y \).
Our notation will be:

\[ \forall x - \text{for every } x \]
\[ \exists y - \text{there exists a } y. \]

Example: All that glitters is not gold.
Let \( G_1(x) - x \) glitters
\( G_2(x) - x \) is gold

Then the intent of the statement can be expressed by:

\[ (\exists x)(G_1(x) \land \neg G_2(x)) \text{ or } \neg (\forall x)(G_1(x) \equiv G_2(x)). \]

Example: Let \( M(x) - x \) is male
\( F(x) - x \) is female
\( P(x,y) - x \) is a parent of \( y. \)

How do we describe an Uncle?

\[ u(x,y) \equiv x \text{ is the uncle of } y \equiv (\forall u)(\forall v)(P(u,x) \land P(u,v) \land M(x) \land P(v,y)). \]

Here are some further examples:

\[ (\forall x)(\exists y)(x < y) - \text{this is true for integers} \]
\[ (\exists y)(\forall x)(x < y) - \text{this is false for integers} \]
\[ (\forall x)(\exists y)(\forall z)(x < y \equiv (x < z) \land (z < y)) - \text{false in integers, true in rationals.} \]
The last example shows that there are things that do not have finite models. In order to express that \( f \) is one-to-one we could write \( \forall x \forall y (\neg(x = y) \Rightarrow \neg(f(x) = f(y))) \). Operations such as multiplication can be expressed as,

\[
\forall x \forall y \forall z \forall w (M(x, y, z) \land (M(x, y, w) \Rightarrow w = z)).
\]

Even concepts such as the continuity of a function \( f \) can now be described:

\[
\forall x \forall \epsilon \forall \delta (\epsilon > 0 \Rightarrow (\delta > 0) \land (0 < |x - x_0|) \land (|x - x_0| < \delta) \Rightarrow |f(x) - f(x_0)| < \epsilon).
\]

A further new item is a constant. For example \( E(4) \) means that 4 is even, or \( \vdash E(4) \) could mean to derive that 4 is even. We are out of Propositional Calculus and moving into what is called First Order Predicate Calculus.

Consider the collection consisting of:

1. A non-empty infinite set \( OV \) called object variables.
2. A set \( F \) with an arity function \( \rho_1 \). We assume \( F_0 = \emptyset \). For \( n > 0 \) let \( F_n = \{ u | u \in F, \rho_1(u) = n \} \). We call \( F_0 \) the set of constants and \( F_n \) for \( n > 0 \) the set of \( n \) place functions. Assume \( F \cap OV = \emptyset \).
3. A set $P$ with an arity function $\rho_2$. Define $P_n = \{v \mid v \in P, \rho_2(v) = n\}$. We assume $P_0 = \emptyset$, $P \neq \emptyset$.

Let $\rho_3:OVUF \to N$ be defined by $\rho_3|F = \rho_1$ and $\rho_3(x) = 0$ for $x \in OV$.

Let $T(OV,F,\rho_1) = \mathcal{F}(OVUF,\rho_3)$ we call $T(OV,F,\rho_1)$ the set of terms.

Let $at(OV,F,P,\rho_1,\rho_2)$ be the sequences of $(OVUFUP)^*$ of the form $S_t, t_n$ where $S \in P_n$ and $t_1, \ldots, t_n \in T$. Note: $T = T(OV,F,\rho_1)$. These are called the atomic formulas.

Examples: $Exy$ can mean $x = y$ where $E \in P_2$

$Efxy4w$ where $f \in F_3$. Informally we write $f(x,y,4) = w$.

4. Let $L = \{\neg, \equiv\}$ and assume $L \cap \text{Nat} = \emptyset$. Let $Q$ be a set which is in a fixed 1-1 correspondence with $OV$. (i.e., take a picture of each $OV$ and call it that.) We assume $Q, L, at, OV$ are pairwise disjoint. Note: We can manufacture a $Q$ easily, for example if $x \in OV$ then $(x,52) \in Q$ etc.

Define $\rho_4 : QULUat \to N$ as follows:

if $w \in at$ then $\rho_4(w) = 0$

if $w \in Q$ then $\rho_4(w) = 1$

if $w \in L$ then $\rho_4(\neg) = 1, \rho_4(\equiv) = 2$. 

44
Let \( PF(OV,F,P,p_1,p_2) = \mathcal{F}(QUVal,p_4) \).

Examples: \( f(x_1,x_2,x_3,x_4) \) is formal notation.
\[
f(g(x_1,x_2,x_3),x_2,y) = fgx_1x_2x_3x_2y
\]
where \( f \) and \( g \) have arity 3.

If \( x \in OV \), let \( Ax \) be its corresponding object in \( Q \). That is, \( Ax \) is a single object related to \( x \). We will not be working with \( Q \) so there will be no confusion.

Another way to state 4 above is:

1. An atomic formula is a predicate formula.
2. If \( A \) and \( B \) are predicate formulas so are \( \neg A \), and \( \Rightarrow AB \).
3. If \( A \) is a predicate formula and \( x \) is an object variable so is \( AxA \) a predicate formula.

A variable is considered free if it is not associated with any quantifier, otherwise it is bound.

For example \( Axu(x) \) is the same as \( Ayu(y) \) and in both cases the variables \( x \) and \( y \) are bound. When we use summation notation in mathematics we have both free and bound variables in the same expression. In \( \sum_{i=1}^{n} f_i \), \( i \) is bound but \( f \) is free. We want to be able to use a
variable as both free and bound in the same formula. A statement with no free variables is a sentence. This means that it has a truth value. In the following example we see a free occurrence of the bound variable x.

\[ u(x) \land \forall x w(x) \quad \text{or} \quad (u(x) \Rightarrow \forall x w(x)) \]

Recall that every \( u \in \text{PF} \) can be written in the form \( u = u_1 u_2 \ldots u_k \) where \( u_i \), \( 1 \leq i \leq k \), satisfies one of the following:

1. \( u_i \in \text{OV} \)
2. \( u_i \in F \)
3. \( u_i \in P \)
4. \( u_i \in \{\neg, =\} \)
5. \( u_i \in \{\forall x | x \in \text{OV}\} \).

The following definitions will be helpful in studying the predicate calculus.

Definition: FV-free variable.

If \( A \) is atomic then \( \text{FV}(A) \) is the set of object variables which occur in \( A \). Example:

\[ \text{FV}(w(3)) = \emptyset \quad \text{if} \quad \rho^2(w) = 1 \quad \text{and} \quad 3 \in F_0. \]

To complete our definition of FV:
FV(\neg A) = FV(A)
FV(A \Rightarrow B) = FV(A) \cup FV(B)
FV(\land x A) = FV(A) \setminus \{x\} \text{ since } x \text{ is bound.}

Note: FV(\land y u(x)) = \{x\}.

AV-all variables
If A is atomic AV(A) = FV(A)
\quad AV(\neg A) = AV(A)
AV(A \Rightarrow B) = AV(A) \cup AV(B)
AV(\land x A) = AV(A) \cup \{x\}.

BV-bound variables
If A is atomic BV(A) = \emptyset
BV(\neg A) = BV(A)
BV(A \Rightarrow B) = BV(A) \cup BV(B)
BV(\land x A) = BV(A) \cup \{x\}.

Examples: A(x,x) or formally Axx. Here occurrences 2 and 3 are free. u(x) = \land x w(x) or formally = u x \land x w x. Here x is bound in occurrences 4 and 6 but free in occurrence 3. Note: \land x is counted as one occurrence.

We use the notation A[V_1, V_2, \ldots, V_k] to mean the free variables of A occur among V_1, \ldots, V_k. We write A[t_1, \ldots, t_k] to be the result of substituting terms t_i for V_i. That is, for every free occurrence of V_i we put the string t_i and so on.
Example: $Uxz$ or $U(x,z)$ and $t_i = fxy$ gives $u(fxy,z)$, where $u \in P_2$ and $f \in F_2$.

It is now our goal to talk about truth in the predicate calculus. We will do proofs in the first order predicate calculus similar to the ones we have done in the propositional calculus.

Notice that we can get into trouble with substitutions. Consider $(Vx)(x=y)$, here $x$ is bound and $y$ is free. This formula is true for rational numbers. Suppose we have the function $+uv$. Substitute $+4x$ for $y$ and obtain $(Vx)(x=+4x)$. This is not true for integers or rationals. We must be careful about substitutions. We restrict substitutions as follows.

Suppose $x$ is free in $A$ and $t$ is a term. We say that $t$ is free for $x$ in $A$ iff for each variable $y$ in $t$, $y$ does not become bound in $A$ if we substitute it for $x$. We will insist that variables do not become bound when we make a substitution.

If $A$ is a formula and $t$ is a term, define $\int_A^x t$ as follows:

$\int_A^x t = A$ if $t$ is not free for $x$ in $A$ and otherwise $\int_A^x t$ is the result of substituting $t$ for $x$ in $A$. 

48
We now list the axioms for predicate calculus.

Axioms for Predicate Calculus

Let $A, B, C$ be formulas. Then the following are axioms.

$P_1$ $A \Rightarrow (B \Rightarrow A)$

$P_2 [(A \Rightarrow (B \Rightarrow C)) \Rightarrow [(A \Rightarrow B) \Rightarrow (A \Rightarrow C)]]$

$P_3 [\neg B \Rightarrow \neg A] \Rightarrow [A \Rightarrow B]$

$P_4$ If $t$ is a term and $t$ is free for $x$ in $A$ then

$$\forall x A[x] \Rightarrow \bigcup_{t} A .$$

$P_5$ If $x$ is not free in $A$ then

$$[\forall x (A \Rightarrow B)] \Rightarrow [A \Rightarrow \forall x B] .$$

Let $\Gamma \subseteq PF$ and let $B \in PF$. We say that $B$ is derivable from $\Gamma$ iff there is a sequence:

$D_1, D_2, \ldots, D_m$ of $PF$ such that

(i) $B = D_m$

(ii) for each $1 \leq i \leq m$ either

(a) $D_i$ is an axiom

(b) $D_i \in \Gamma$

(c) for each $p, q < i, p \neq q, D_q = D_p \Rightarrow D_i$

(d) for some $p < i D_i = \forall x D_p$ where $x$ is some variable. (generalization)

Example: If $P(x) \Rightarrow P(x)$ is true then $\forall x (P(x) \Rightarrow P(x))$. This gets us from the metalinguistic language of the inference to our object language.
Recall, the [ ] are descriptive notation to indicate the free variables in a formula. \( A[x,y] \) means all the free variables of \( A \) appear among \( X \) and \( Y \).

Suppose \( Y \) is free for \( x \) in \( A[x] \), then

\[
A[x] \vdash \exists^x A. \quad \text{Note: we cannot say } A[x] \vdash \exists^x A \Rightarrow A[y] \Rightarrow A[x].
\]

PROOF: 1. \( A[x] \) by (ii)(b)

2. \( \forall xA[x] \) generalization

3. \( \forall xA[x] \Rightarrow \exists^x A \) by \( P_4 \)

4. \( \exists^x A \) by detachment.

Any result of substitution of PF in a propositional tautology yields a theorem. Recall a theorem is derivable, \( \vdash A \) then \( A \) is a theorem. If \( A \) is an instance of a propositional tautology then \( \vdash A \).

Definitions:

\[
\begin{align*}
VAB &: = \neg AB \equiv AVB &: = \neg A \land B \\
\wedge AB &: = \neg (A \land B) \equiv A \lor B &: = A \lor (B \lor A) \\
\leftrightarrow AB &: = A \rightarrow B \land B \rightarrow A \equiv (A \rightarrow B \lor B \rightarrow A)
\end{align*}
\]

\( VxA \) := \( \neg \exists^x A \) this means there exists. The rules for binding are the same then as for \( \forall x \).
A term is called closed iff it has no object variables. A predicate formula is called closed or a sentence iff it has no free variables. A formula \( A \) is said to be in prenex form iff \( A = Q_1, Q_2, \ldots, Q_r B \), where each \( Q_i \) has the form \( \forall x \) or \( \exists x \) and every variable in \( B \) is free. This means that \( B \) is a quantifier free formula.

We want to be able to change the names of bound variables. Example \( \int_0^1 x \, dx \) is the same as \( \int_0^1 y \, dy \).

Definition: Let \( A, B \in PF \) and let \( x, y \in \) object variables where \( X \neq Y \). Suppose \( x \) is free in \( A \), \( Y \) is free for \( x \) in \( A \) and \( Y \) is not free in \( A \). Then we say that \( A \) and \( B \) are similar iff \( B \equiv A \).

Notice that \( Y \) is free in \( B \). The condition is a symmetrical one. i.e., \( X \) is free for \( Y \) in \( B \) and \( x \) is not free in \( B \).

Theorem: If \( A \) and \( B \) are similar where \( B = \int_y^X A \) then \( \vdash \forall x A \equiv \forall Y B \).

PROOF: 1. \( \forall x A = \int_y^X A \) by P4 i.e., \( \forall x A \equiv B \)

2. \( \forall Y (\forall x A \equiv B) \) generalization. Note: \( Y \) is not free in \( \forall x A \)

3. Hence \( \forall Y (\forall x A \equiv B) \equiv (\forall x A \equiv \forall Y B) \) by P5

4. \( \forall x A \equiv \forall Y B \) by detachment.
Corollary: With the same hypothesis $\Lambda x A \leftrightarrow \Lambda y B$.

PROOF: The proof follows from the symmetry of the definition above and the tautology $u \Rightarrow (\forall = u \Lambda V)$.

Note: If $Y$ is free for $x$ in $A$ then $A \vdash \begin{array}[]{c} x \\ y \\ A \end{array}$

but $\vdash A \Rightarrow \begin{array}[]{c} x \\ y \\ A \end{array}$ does not follow.

Example: $A \equiv x$ is even. Note: $x$ is free in $A$.

\begin{array}[]{c} x \\ y \\ A \equiv y \text{ is even} \\ y \\ A \Rightarrow \begin{array}[]{c} x \\ y \\ A \equiv \Lambda x A Y \text{ (x is even } \Rightarrow \text{ Y is even)} \\ y \\ \Lambda Y \text{ (2 is even } \Rightarrow \text{ Y is even)} \\ y \\ (2 \text{ is even } \Rightarrow \text{ 3 is even).} \end{array} \end{array} \end{array}

Suppose $\Gamma, A \vdash B$. Let $D_1, \ldots, D_m$ be the derivation of this deduction. For each $j = 1, \ldots, m$ we say that $D_j$ depends on $A$ iff either

1) $D_j = A$

2) (a) for some $p, q < j$, $D_q = D_p \Rightarrow D_j$ where $D_p$ depends on $A$

(b) for some $p < j$, $D_j = \Lambda x D_p$ where $D_p$ depends on $A$.

Deduction Theorem.

Suppose $\Gamma, A \vdash B$ and no step of the derivation is of the form $\Lambda x C$ where $C$ depends on $A$, $x$ is free in $A$ and $C$ is an earlier step. Then $\Gamma \vdash A \Rightarrow B$. 52
Corollary. If $\Gamma, A \vdash B$ and $A$ is closed then $\Gamma \vdash \lambda = B$.

PROOF: of Deduction Theorem

Let $D_1, \ldots, D_m$ be the derivation of $\Gamma, A \vdash B$ satisfying the hypothesis. We show that $\Gamma \vdash A \Rightarrow D_j$ for $j = 1, \ldots, m$. That $\Gamma \vdash A \Rightarrow D_1$ is as before.

Suppose for some $k$ with $1 < k \leq m$ we have that $\Gamma \vdash A \Rightarrow D_j$ for $1 \leq j < k$.

If $D_k$ is obtained from the earlier steps using detachments then the proof is as before. That is to say we have no quantifiers to worry about as in the propositional calculus. Hence we may assume that $D_k = \lambda \mathcal{A}_p$ for some $p < k$.

There are two cases to consider.

(1) $D_p$ does not depend on $A$.
(2) $D_p$ does depend on $A$.

For (1) the derivation of $D_p$ can be so arranged that $\Gamma \vdash D_p$. We may extend the derivation as follows:

\begin{align*}
\lambda \mathcal{A}_p & \quad \text{generalization} \\
\lambda \mathcal{A}_p & \Rightarrow (A \Rightarrow \lambda \mathcal{A}_p) \quad \text{by } P_1 \\
A & \Rightarrow \lambda \mathcal{A}_p \quad \text{detachment} \\
\text{Hence, } \Gamma \vdash A \Rightarrow D_k
\end{align*}

For (2) in this case $x$ is not free in $A$. Since $\Gamma \vdash A \Rightarrow D_p$ from $p < k$ now add
\[ \Lambda x(\Lambda x D_p) \text{ generalization} \]
\[ \Lambda x(\Lambda x D_p) = (\Lambda x \Lambda x D_p) \text{ by } P_5 \]
\[ A = \Lambda x D_p \text{ so } \Gamma \vdash A = D_k. \]

Note: \( x \) is not free in \( A \) in case (2) due to the hypothesis. In either case we have that \( \Gamma \vdash A = D_k \) so by induction we have \( \Gamma \vdash A = B. \)

The following theorem shows that the quantifier has no affect if it does not contain a variable which is free in the quantified formula.

Theorem: If \( x \) is not free in \( A \), then \( \Lambda x A \iff A. \)

PROOF: 1. \( \Lambda x A = \int X A \) by \( P_4 \). Note: \( \int X A = A \)

2. \( \Lambda x A = A \)

3. \( A \vdash \Lambda x A \) using generalization

4. \( \vdash A = \Lambda x A \) by the deduction theorem since \( x \) is not free in \( A. \)

Hence we have both \( \Lambda x A = A \) and \( A = \Lambda x A. \)

Another way to establish that \( A = \Lambda x A \) follows:

1. \( A = A \) tautology

2. \( \Lambda x (A = A) \) by generalization

3. \( \Lambda x (A = A) = (A = \Lambda x A) \) since \( x \) is not free in \( A \)

4. \( A = \Lambda x A \) by detachment.
We will now list some propositional tautologies that will be useful. They can be established by truth tables or formal derivation.

Law of Substitution of Equivalence

\[(A<->B) \land (C<->D) \Rightarrow (A\land C)<=>(B\lor D)]
\[(AVC)<=>(BVD)]
\[(A\lor C)<=>(B\land D)]

\[A<=>B \iff [A=\Rightarrow B]

We have established that if \( x \) is free in \( A \) and \( Y \) is not free in \( A \) but \( Y \) is free for \( x \) in \( A \) then \( \forall x A \iff \forall y B \) where \( B = \int x A \). Now, applying the substitution tautology

\[\forall x A \lor \forall x B \iff \forall u \int x A \lor \forall x B \text{ where } u \text{ is a variable not in } A.

The semantics of predicate calculus needs to be established. Let \( L \) be a predicate calculus. Notation: \( P \)-predicate, \( F \)-function, \( L = \{OV,P,F\} \)

An \( L \)-structure for \( L \) is a system consisting of:
1. A set \( A \) called the universe of the \( L \)-structure, \( A \neq \emptyset \).
2. For each function \( C \) of arity 0, the assignment of some fixed member \( c \in A \).
3. For each function \( f \) of arity \( n > 0 \) the assignment of a fixed function \( \overline{f} : A^n \to A \).

4. For each predicate \( Q \) of arity \( n > 0 \) we assign a function \( \overline{Q} : A^n \to \{0,1\} \).

We write \( u = \{A,P,F\} \).

A valuation of \( L \) relative to \( u \) is a function \( S : OV \to A \).

1. If \( C \in F_0 \) define \( s(C) = \overline{C} \).

2. If \( t_1, \ldots, t_n \in F_n \) define \( s(t_1, \ldots, t_n) = \overline{f}(s(t_1), \ldots, s(t_n)) \) in \( A \).

3. If \( B \) is an atomic formula, \( B = C t_1, \ldots, t_n \) where \( C \in P_n \) define \( s(B) = \overline{Q}(s(t_1), \ldots, s(t_n)) = \{0 \}

4. If \( B \in PF \) (a) \( B = \overline{\neg}D \), \( s(B) = 1 - s(D) \)
   (b) \( B = (D \overline{\land} C) \), \( s(B) = \max(s(\overline{\neg}D),s(C)) \).

Definition: If \( s \) is a valuation on \( OV \), \( x \in OV \) and \( a \in A \) define \( s(a/x) \) to be the valuation \( s' \) defined by \( s'(y) = s(y) \) for \( y \neq x \), \( y \in OV \), and \( s'(x) = a \).

\[
s(a/x)(y) = \begin{cases} 
  s(y) & y \neq x \\
  a & y = x 
\end{cases}
\]

If \( D = A \times B \) then take \( s(D) = \inf(s(a/x)(B)|a \in A) \).

i.e., we get zero if it fails for any one of them and otherwise one. We say that \( s \) satisfies \( D \) iff \( s(D) = 1 \). (Alternatively \( s \) is a model for \( D \).)
We define \( \models_u D \) to mean that every \( s \) obtained from \( u \) satisfies \( D \). (read \( \models_u D \), \( D \) is valid relative to \( u \).) We say that \( D \) is valid iff \( \models_u D \) for every \( L \)-structure \( u \). The notation then is \( \models D \).

It is now time to consider, what is the relationship between validity and deducibility? This is the heart of model theory.

If \( \Gamma \subseteq PF \) and \( D \in PF \) we write \( \Gamma \models_u D \) iff there is no valuation \( s \) in \( u \) such that \( s(D) = 0 \) and \( s(B) = 1 \) for all \( B \in \Gamma \). We see that \( \models_u D \) is the same as \( \phi \models_u D \). If \( \Gamma \subseteq PF \) and \( s \) is a valuation in \( u \), then \( s \) satisfies \( \Gamma \) iff \( s(B) = 1 \) for all \( B \in \Gamma \). We also say that \( s \) is a model for \( \Gamma \). We write \( \Gamma \models D \) iff \( \Gamma \models_u D \) for all \( u \).

**Theorem:** If \( \Gamma \subseteq PF \), \( D \in PF \) and \( \Gamma \models D \) then \( \Gamma \models D \).

**PROOF:** Let \( u \) be an \( L \)-structure. First we note that for each axiom \( D \), \( s(D) = 1 \). \( P1-P3 \) are tautologies and hence offer no difficulty. For \( P4 \), let \( s \) be a valuation and \( AxD = \int^x \) be an example of axiom \( P4 \).

If \( s(AxD) = \int^x \) then \( s(AxD) = 1 \) and \( s(\int^x D) = 0 \).

Let \( s(t) = a \). Then let \( s' = s(a/x) \). Now we see that \( s'(D) = 0 \). Hence, by definition \( s(AxD) = 0 \) which is a contradiction since \( s(AxD) = 1 \) means that
s(a/x)(D) = 1 for all a ∈ A. Now, for P5, let
c = Ax(D=B) = (D=∧xB) be an instance of P5. Recall
that x is not free in D. Let s be a valuation.
If s(c) = 0 then s(Ax(D=B)) = 1 and s(D=∧xB) = 0.
Hence, s(D) = 1 and s(∧xB) = 0. Thus for some
a ∈ A and s' = s(a/x) we have s'(B) = 0. Note,
s'(D) = 1, since x is not free in D, and s'(D=B) = 0.
Hence, s(Ax(D=B)) = 0 which is a contradiction.

Note: detachment preserves satisfiability, for
if s(D) = 1 and s(D=B) = 1 then s(B) = 1.

For generalization we must be more careful.
Suppose B is satisfied by every valuation which
satisfies Γ. Then every valuation which satisfies
Γ must satisfy ∧xB. For if s is a valuation which
satisfies Γ but s(∧xB) = 0, then there is an a ∈ A
such that if s' = s(a/x) then s'(B) = 0. But if s'
satisfies Γ then s'(B) = 1. Hence, s(∧xB) = 1.
The problem here is that we don't know if s'
satisfies Γ. If Γ is a set of closed formulas then we know
that s' satisfies Γ since x would not be free in
any D ∈ Γ.

Because there is a problem in the generalization
part of the previous proof, I will state the theorem in
a more restricted manner. The previous proof then
follows with the generalization part as given here.
Theorem: Let \( \Gamma \) be a set of closed formulas. If \( \Gamma \vdash D \) then \( \Gamma \models D \).

Generalization part. Suppose \( B \) is satisfied by every valuation which satisfies \( \Gamma \) and then by generalization \( \Gamma \vdash \forall x B \). If we do not have \( \Gamma \models \forall x B \) then there is a valuation \( s \) such that \( s(D) = 1 \) for all \( D \in \Gamma \) and \( s(\forall x B) = 0 \). Hence, we can find an \( a \in A \) such that if \( s' = s(a/x) \) then \( s'(B) = 0 \).

But \( s'(D) = 1 \) for all \( D \in \Gamma \) which gives us a contradiction. Hence, \( \Gamma \models \forall x B \).

Theorem: (1) \( \vdash \forall x (A \Rightarrow B) \Rightarrow (\forall x A \Rightarrow \forall x B) \)

(2) \( \vdash \) If \( x \) is not free in \( A \) then
(a) \( \forall x (A \Rightarrow B) \iff (\forall A \Rightarrow \forall x B) \)
(b) \( \forall x (B \Rightarrow A) \iff (\forall B \Rightarrow \forall x A) \).

PROOF: (1) We first show that \( \forall x (A \Rightarrow B), \forall x A \vdash \forall x B \)
1. \( \forall x (A \Rightarrow B) \Rightarrow (A \Rightarrow B) \) by P4 using \( \int x D = D \)
2. \( \forall x (A \Rightarrow B) \) from \( \Gamma \)
3. \( A \Rightarrow B \) detachment
4. \( \forall x A \) from \( \Gamma \)
5. \( \forall x A \Rightarrow A \) by P4 using \( \int x D = D \)
6. \( A \) detachment
7. \( B \) detachment
8. \( \forall x B \) generalization.
Hence $\forall x(A \Rightarrow B) \mid - (\forall x A \Rightarrow \forall x B)$ using the deduction theorem which is valid since $x$ is not free in $\Gamma$. With one more application of the deduction theorem we have our result. $\forall x(A \Rightarrow B) \Rightarrow (\forall x A \Rightarrow \forall x B)$.

Note: The following special case of the deduction Theorem. If $\Gamma, B \mid - C$ is a deduction in which no variable free in $\Gamma \cup \{B\}$ has had generalization applied to it, then $\Gamma \mid - B \Rightarrow C$.

(2)(a) $\forall x (A \Rightarrow B) \Rightarrow (A \Rightarrow \forall x B)$ is an axiom. We show $A \Rightarrow \forall x B \mid - \forall x (A \Rightarrow B)$

1. $\forall x B \Rightarrow B$ by P4
2. $A \Rightarrow \forall x B$ from $\Gamma$
3. $A = B$ using $(u \Rightarrow v) = ((u \Rightarrow w) \Rightarrow (u \Rightarrow w))$
4. $\forall x (A \Rightarrow B)$ generalization, recall $x$ is not free in $A$
5. Hence $(A \Rightarrow \forall x B) \Rightarrow \forall x (A \Rightarrow B)$ by deduction.

(b) $\forall x (B \Rightarrow A) \Rightarrow (\forall x B \Rightarrow A)$

1. $(B \Rightarrow A) \Rightarrow (\neg A \Rightarrow \neg B)$ by T8.5
2. $\forall x [(B \Rightarrow A) \Rightarrow (\neg A \Rightarrow \neg B)]$ generalization
3. $\forall x (B \Rightarrow A) \Rightarrow \forall x (\neg A \Rightarrow \neg B)$ using part (1)
4. $\forall x (\neg A \Rightarrow \neg B) \Rightarrow (\neg A \Rightarrow \forall x (\neg B))$ P4
5. $\forall x (B \Rightarrow A) \Rightarrow (\neg A \Rightarrow \forall x (\neg B))$ transitivity
6. $(\neg A \Rightarrow \forall x (\neg B)) \Rightarrow (\neg \forall x (\neg B) \Rightarrow A)$
7. \((-A\supset \lambda x(\neg B)) \equiv (VxB \supset A)\) by definition of VxB
8. \(\lambda x(B \supset A) \equiv (VxB \supset A)\) transitivity

Note: This argument is reversible.

Now the other way: \((VxB \supset A) \equiv \lambda x(B \supset A)\)

1. \((-\lambda x(\neg B) \supset A) \equiv (-\lambda x(\lambda x(\neg B)) \supset A)\) by T8.5
2. \((-\lambda x(\neg B)) \equiv \lambda x(\neg A \supset \neg B)\) by 2(a)
3. \((-\lambda x(\neg B)) \equiv (B \supset A)\)
4. \(\lambda x((-A \supset \neg B) \equiv (B \supset A))\) generalization
5. \(\lambda x((-A \supset \neg B) \equiv (B \supset A)) \equiv \lambda x(-A \supset \neg B) \equiv \lambda x(B \supset A)\)
   by (1) of this theorem
6. \(\lambda x(-A \supset \neg B) \equiv \lambda x(B \supset A)\) detachment
7. \((VxB \supset A) \equiv \lambda x(B \supset A)\) transitivity on 1,2,6.

Theorem: If \(x\) is not free in \(B\) then:

1. \(\lambda x(A\supset B) \iff (\lambda xA\supset B)\)
2. \(\lambda x(A\vee B) \iff (\lambda xA\vee B)\)
3. \(Vx(A\vee B) \iff (VxA\vee B)\)
4. \(Vx(A\supset B) \iff (VxA\supset B)\).

We will also need the following theorems. First here is an informal argument for \(Vx(A \supset B), \lambda xA \vdash VxB\).

We pick an \(a\) such that \(\int_a^x (A \supset B)\) is true so \(\int_a^x A \supset \int_a^x B\).
Now, $\Lambda x A$ means $\int x A$ is true so $\int x B$ is true. Hence, $V x B$. We now give a formal proof.

Theorem: $V x (A \Rightarrow B)$, $\Lambda x A \vdash V x B$. We prove as an intermediate stage: $\Lambda x A, \Lambda x (\neg B) \vdash \Lambda x (\neg (A \Rightarrow B))$.

1. $\Lambda x A$ by $\Gamma$
2. $\Lambda x (\neg B)$ by $\Gamma$
3. $\Lambda x A \Rightarrow A$ by P4
4. $A$ detachment
5. $\Lambda x (\neg B) \Rightarrow \neg B$ by P4
6. $\neg B$ detachment
7. $A \Rightarrow (\neg B \Rightarrow (A \Rightarrow B))$ tautology
8. $\neg (A \Rightarrow B)$ detachment
9. $\Lambda x \neg (A \Rightarrow B)$ generalization

hence $\Lambda x A \vdash \Lambda x (\neg B) \Rightarrow \Lambda x \neg (A \Rightarrow B)$ deduction

$\Lambda x A \vdash \neg \Lambda x (\neg (A \Rightarrow B)) \Rightarrow \neg \Lambda x (\neg B)$

$\Lambda x A \vdash V x (A \Rightarrow B) \Rightarrow V x B$

$\Lambda x A, V x (A \Rightarrow B) \vdash V x B$.

Now, recall if $x$ is not free in $A$

1. $\Lambda x (A \Rightarrow B) \iff (A \Rightarrow \Lambda x B)$
2. $\Lambda x (B \Rightarrow A) \iff (V x B \Rightarrow A)$.

We now add to these:

3. $V x (A \Rightarrow B) \iff (A \Rightarrow V x B)$
4. $V x (B \Rightarrow A) \iff (\Lambda x B \Rightarrow A)$.
PROOF: To use in 3, we first show that if \( x \) is not free in \( A \) then \( \forall x A \iff A \).

1. \( x \) is not free in \( \neg A \) since \( x \) is not free in \( A \).

2. \( \forall x (\neg A) \iff \neg A \)

3. \( \neg \forall x (\neg A) \iff \neg \neg A \) from 2 to 3 using
   \((u\iff v) \iff (\neg u\iff \neg v)\)

4. \( \forall x A \iff A \) since \( \neg \forall x (\neg A) \iff \forall x A \) and \( \neg \neg A \iff A \).

Before doing 3, we note \( \neg (u \iff v) \iff u \) and \( \neg (u \iff v) \iff v \) are tautologies. In fact these are the same as \( u \lor v \iff u \) and \( u \land v \iff v \).

We also need \( [u \iff (v \iff (w \iff w))] \iff [u \iff v] \) which is the logical form of a contradiction. It is also a tautology. Now, for the proof of 3.

a) \( \forall x (A \rightarrow B) \iff (A \rightarrow \forall x B) \)

1. \( \forall x A \iff A \) by P4

2. \( \forall x (A \rightarrow B), \forall x A \vdash \forall x B \) theorem

3. \( \forall x (A \rightarrow B), A \vdash \forall x B \)

4. \( \forall x (A \rightarrow B) \vdash A \rightarrow \forall x B \) deduction

5. \( \forall x (A \rightarrow B) \vdash (A \rightarrow \forall x B) \) deduction.

Now, to go the other way:

b) \( (A \rightarrow \forall x B) \rightarrow \forall x (A \rightarrow B) \)

1. \( (A \rightarrow \forall x B), \forall x (A \rightarrow B) \vdash (w \equiv w) \)

2. \( \neg \forall x (A \rightarrow B) \) in \( \Gamma \)

63
3. \( \forall x(\neg(A\Rightarrow B)) \) by definition of \( \forall x(A\Rightarrow B) \)
4. \( \neg(A\Rightarrow B) \) by P4
5. \( \neg(A\Rightarrow B) = A \) tautology
6. \( A = \forall xB \) in \( \Gamma \)
7. \( \forall xB \) detachment
8. \( \neg(A\Rightarrow B) = \neg B \) tautology
9. \( \neg B \) detachment
10. \( \forall x(\neg B) \) generalization
11. \( \neg(\forall x(\neg B)) \) theorem
12. \( \forall xB \) by definition of \( \forall xB \) and 11.
13. \( \forall xB = (\forall xB = \neg(\forall xB = \forall xB)) \) tautology
14. \( A = \forall xB \mid (\forall x(A\Rightarrow B) = \neg(w\Rightarrow w)) \) deduction
15. \( A = \forall xB \mid ((w\Rightarrow w) = \forall x(A\Rightarrow B)) \) theorem
16. \( ((w\Rightarrow w) = \forall x(A\Rightarrow B)) = \forall x(A\Rightarrow B) \) theorem
17. \( A = \forall xB \mid \forall x(A\Rightarrow B) \)
18. \( (A\Rightarrow \forall xB) = \forall x(A\Rightarrow B) \)

We will now prove 4 which says \( \forall x(B\Rightarrow A) \Leftrightarrow (\forall xB\Rightarrow A) \).

a) \( \forall x(B\Rightarrow A) = (\forall xB\Rightarrow A) \)

recall \( \forall x(B\Rightarrow A), \forall xB \mid \forall xA \) we did already.

\( \forall xA \Leftrightarrow A \) since \( x \) is not free in \( A \) so

\( \forall x(B\Rightarrow A), \forall xB \mid A \) so by using deduction

\( \forall x(B\Rightarrow A) = (\forall xB\Rightarrow A) \)

b) Now for \( (\forall xB\Rightarrow A) = \forall x(B\Rightarrow A) \)

\( \forall xB = A, \forall x(B\Rightarrow A) \mid \neg (w\Rightarrow w) \)
1. $\neg Vx(B\Rightarrow A)$ assumption
2. $\forall x(\neg (B\Rightarrow A))$ by definition of $Vx$
3. $\neg (B\Rightarrow A) \Rightarrow \neg A$ tautology
4. $\neg (B\Rightarrow A) \Rightarrow B$ tautology
5. $\forall x(\neg (B\Rightarrow A) \Rightarrow \neg A)$ generalization
6. $\forall x \neg (B\Rightarrow A) \Rightarrow \forall x \neg A$ theorem
7. $\forall x \neg A$ detachment
8. $\neg A$ P4
9. $\forall x(\neg (B\Rightarrow A) \Rightarrow B)$ generalization
10. $\forall x \neg (B\Rightarrow A) \Rightarrow \forall x B$ theorem
11. $\forall x B$ detachment
12. $\forall x B \Rightarrow A$ assumption
13. $A$ detachment
14. $A \Rightarrow (\neg A \Rightarrow \neg (A\Rightarrow A))$ tautology
15. $\neg (A\Rightarrow A)$ by two detachments
16. Hence, $\forall x B \Rightarrow A \mid \neg Vx(B\Rightarrow A)$.

Note: We did not use generalization on any formula in which $x$ is free. Therefore, $(\forall x B\Rightarrow A) \Rightarrow Vx(B\Rightarrow A)$.

For the work that is to follow we will need certain results from ordinality theory and cardinality theory. Ordinality theory is easier to deal with technically but cardinality theory is easier to understand. We will assume Zorn's lemma. The following definitions are needed.
Definition: A strict partial order, $<$, on a set $X$ is a well-ordering iff:

1. it is a total order, i.e., given $x, y$ one and only one of the following holds: $x < y$, $x = y$, $y < x$
2. if $A \subseteq X$ is a non-empty subset then $A$ has a least element, i.e., there is an $x \in A$ such that if $y \in A$ and $y \neq x$ then $x < y$.

Note: 1 and 2 above imply each other.

Here is an example of a set with a well-ordering. Consider $N = \{0, 1, 2, 3, \ldots \}$ with $\lt$.

We state the following theorem without proof.

Theorem: Zorn's axiom implies that every non-empty set has a well-ordering in it. If it is an infinite set there are infinitely many well-orderings in it.

Consider the following well-ordering. The even integers $0, 2, 4, \ldots$ under their normal ordering and the odd integers $1, 3, 5, \ldots$ under their normal ordering given the following ordering, $0, 2, 4, \ldots < 1, 3, 5, \ldots$. That is to say we consider every even integer less than every odd integer. We see now that the number one has infinitely many elements of $N$ less than it and yet there is no element of $N$ that is the next one down from one.
Another interesting situation arises if we consider mapping 1 to 1-1, 2 to 1 - \( \frac{1}{2} \), 3 to 1 - \( \frac{1}{3} \), etc.

Here is the graph

```
1 ---- 2 ---- 3 ---- 4 ---- etc.
```

we see the points approaching a particular point. The same thing could be done mapping the whole numbers toward the \( \frac{1}{2} \) or the 1 - \( \frac{1}{3} \) or the 1 - \( \frac{1}{4} \) etc. The heavy dots look like the integers from the point of view of order. Each set in between is the same in this respect.

There is a difference in the heavy dots and the intermediate tick marks, however. To see this we give the following definitions.

Definition: Let \((X, <)\) be a well-ordered set and \(x, y \in X\). We say that \(y\) is the immediate successor of \(x\) iff \(x < y\) and there is no \(z\) such that \(x < z\) and \(z < y\). If \(x\) is not the least element of \(X\) and \(x\) is not the immediate successor of any element then we will call \(x\) a limit element in the order \(<\).

Let \((X, <)\) be a well-ordered set and \(P\) a predicate having one free variable, \(u\). If \(x \in X\) assume that \(P(x) = \{0\}\) where \(P(x)\) means the substitution of \(u^x\) and then finding the value. The next theorem gives us mathematical induction.
Theorem: Let $x_0$ be the least element of $X$. If $P(x_0) = 1$ and for each $y \in X$, if $P(z) = 1$ for all $z < y$ it follows that $P(y) = 1$, then for all $x \in X$, $P(x) = 1$.

We can state this theorem as:

1. $P(x_0) = 1$
2. $\forall y \forall z ((x_0 < y) \land (z < y) \land ((P(z) = 1) \implies (P(y) = 1)))$. Then $\forall x (P(x) = 1)$.

Recall that Zorn's Lemma implies that every set can be well-ordered. We want to show that the set of well-orderings can be well-ordered. To do this we need the following theorem.

Theorem: Let $(X, \prec)$ and $(Y, \prec)$ be well-ordered sets. Then one and only one of the following three possibilities hold:

(a) There is a one-to-one onto function $f : X \rightarrow Y$ such that $\forall x \forall y (x \in X \land y \in X \land x < y \implies f(x) < f(y))$.
(b) There is a function $f : X \rightarrow Y$ which is one-to-one and not onto such that $\forall x \forall y (x \in X \land y \in X \land x < y \implies f(x) < f(y))$ and there is a $z \in Y$ such that $(f(x) \mid x \in X) = \{u \mid u \in Y \land u < z\}$. This can be stated in the logical notation as:

$\forall z (z \in Y \land (\exists x (x \in X \land f(x) < z))) \land (\forall u (u \in Y \land u < z \implies \forall x (x \in X \land f(x) = u)))$. 

68
(c) There is a function \( f : Y \rightarrow X \) which is one-to-one and not onto like (b) with \( X \) and \( Y \) interchanged.

PROOF: Suppose (a) and (c) do not hold. Let \( x_0 \) be the least element of \( X \) and \( u_0 \) the least element of \( Y \). Let \( f(x_0) = u_0 \). Let \( x' \in X \) and suppose that for all \( x < x' \), \( f(x) \) is defined and satisfies \( f(x) < f(y) \) where \( x < y \) and if \( u < f(x) \) for some \( x < x' \) then there is an \( x_1 < x' \) such that \( f(x_1) = u \). Note, that this condition holds for \( x' = x_0 \). Let \( A \subseteq Y \) be defined by \( A = \{ f(x) | x < x' \} \). Clearly \( A \neq Y \) or else (c) holds using \( f \) inverse.

Let \( B = Y \setminus A \). Then \( B \neq \emptyset \). Let \( w \in B \) be the least element of \( B \). if \( w < f(x) \) for some \( x < x' \) then \( w = f(x_1) \) for some \( x_1 < x' \) by hypothesis. Hence, we would have \( w \in A \) which is a contradiction. Thus \( f(x) < w \) for all \( x < x' \). Let \( f(x') = w \). To see that the hypothesis is satisfied note that \( A = \{ u | u < w, u \in Y \} \). Thus, \( f \) is defined on all of \( X \) and if \( C = \{ f(x) | x \in X \} \) then \( z \) is the least element of \( Y \setminus C \). (\( C \neq Y \) or else (a) holds.)

Here are some simple facts without proof that come from the previous theorem. First we need the following:
Definition: Ordinal numbers. An ordinal number is a set $\alpha$ such that:

(i) if $\alpha \neq \emptyset$ then $\emptyset \in \alpha$

(ii) if $y \in \alpha$ and $x \in Y$ then $x \in \alpha$. In other words, transitive under $\epsilon$.

(iii) $\epsilon$ well orders $\alpha$.

We will now construct such a set,

$\emptyset \equiv 0$

$\{\emptyset\} \equiv \{0\} \equiv 1$

$\{\emptyset, \{\emptyset\}\} \equiv \{0,1\} \equiv 2$

$\{\emptyset, \{\emptyset\}, \{\{\emptyset\}\}\} \equiv \{0,1,2\} \equiv 3$ etc.

etc.

$w = \{0,1,2,\ldots,n,\ldots\}$

$w+1 = \{0,1,2,\ldots,n,\ldots\} \cup \{w\} \equiv w \cup \{w\}$.

So for any ordinal $\alpha$ we have $\alpha = \{\gamma | \gamma \in \alpha\}$ and $\alpha + 1 = \alpha \cup \{\alpha\}$. Our theorem says that if we take a well-ordered set there is a mapping into a set of ordinals. There is a unique ordering that is onto. The last ordinal that is one-to-one with a set is a very important one. It is the cardinality of the set.

Given any set $X$ there is an ordinal number $\gamma$, so that we can write $X = \{x_\alpha | \alpha < \gamma\} x_\alpha \in X$. (This is another version of the axiom of choice.) In particular, $\gamma$ can be chosen so that no smaller $\gamma$ will do. This least $\gamma$ is called the cardinal number of $X$. 70
To show that not all ordinals are cardinals consider \( w + 1 \). Any set that can be indexed by \( w + 1 \) could be indexed by \( w \) so \( w + 1 \) is not a cardinal. Here is an example: \( x_0, x_1, x_2, \ldots, x_n, \ldots \) can be placed one-to-one with the set \( \{ x_j | j < w \} \). We could now use the following map \( x_1 + y_0, x_2 + y_1, \ldots, x_n + y_{n-1}, \ldots, x_0 + y_w \). Now, \( X = \{ y_\alpha | \alpha < w+1 \} \) so \( w + 1 \) can't be a cardinal since the set could be indexed by \( w \).

If \( X \) is an infinite set and \( X^F \) is the set of all finite subsets of \( X \) then \( X \) and \( X^F \) have the same cardinal.

If the set of all basic symbols in a predicate calculus has cardinality \( \alpha \), then the cardinality of the predicate formulas is \( \alpha \) and also the cardinality of the set of closed predicate formulas is \( \alpha \). (Basic symbols are variables, constants, predicate symbols, and function symbols.) Note, a formula corresponds to a finite subset of the basic symbols with possible repeats.) Everything we use in our predicate calculus is countable.

We refer to the cardinality of the set of basic symbols of a predicate calculus as the cardinality of the predicate calculus.

The following theorem will be needed.
Theorem: Let \( \Gamma \) be a set of closed formulas and let \( A \) be a closed formula. If \( \Gamma \) is consistent and it is not the case that \( \Gamma \vdash \neg A \), then \( \Gamma \cup \{ A \} \) is consistent.

PROOF: Suppose \( \Gamma \cup \{ A \} \) is not consistent, then there is a formula \( B \) such that \( \Gamma, A \vdash B \) and \( \Gamma, A \vdash \neg B \).

Then \( \Gamma \vdash A \Rightarrow B \) and \( \Gamma \vdash A \Rightarrow \neg B \). Using the following tautology \( (A \Rightarrow B) \Rightarrow ((A \Rightarrow C) \Rightarrow (A \Rightarrow (B \Rightarrow C))) \) where we let \( C = \neg B \) we have \( \Gamma \vdash A \Rightarrow (B \Rightarrow \neg B) \) and hence \( \Gamma \vdash \neg A \) which is a contradiction of the hypothesis. So \( \Gamma \cup \{ A \} \) is consistent.

Theorem: Let \( \Gamma \) be a consistent set of closed formulas. Then there is a consistent set \( \Gamma' \) of closed formulas such that \( \Gamma \subseteq \Gamma' \) and \( \Gamma' \) is maximal relative to \( \subseteq \) and consistency. That is if we added any formula to \( \Gamma' \) the new set of formulas would be inconsistent.

PROOF: Let \( \gamma \) be the cardinality of the predicate calculus. We may write the set of closed formulas as \( \{ A_\alpha | \alpha < \gamma \} \). We construct sets \( \Gamma_\alpha, \alpha < \gamma \) with the following properties:

(a) \( \Gamma_\alpha \) is consistent

(b) if \( \alpha < \alpha' \) then \( \Gamma_\alpha \subseteq \Gamma_{\alpha'} \).
Note: the parameter $\alpha$ lets us have infinitely many permutations. Let $\Gamma' = \bigcup \{ \Gamma_\alpha \mid \alpha < \beta \}$ then we observe that $\Gamma'$ is consistent.

For suppose there is a formula $B$ such that $\Gamma' \vdash B$ and $\Gamma' \not\vdash \neg B$. Then let $D_1, \ldots, D_m$ be the derivation of $B$ and $E_1, \ldots, E_n$ be the derivation of $\neg B$. Let $F_1, \ldots, F_k$ be the formulas of $\Gamma'$ appearing in $(D_1, \ldots, D_m) \cup (E_1, \ldots, E_n)$. Hence $(F_1, \ldots, F_k) \vdash B$ and $(F_1, \ldots, F_k) \vdash \neg B$. For each $j = 1, \ldots, k$ we can find $\alpha_j$ such that $\alpha_j < \beta$ and $F_j \in \Gamma_\alpha_j$. Since the ordinals are totally ordered we may write

$$\alpha_{j_1} \leq \alpha_{j_2} \leq \ldots \leq \alpha_{j_k}$$

where $j_1, \ldots, j_k$ is some rearrangement of $1, \ldots, k$. Thus since $\Gamma_{\alpha_{j_p}} \subseteq \Gamma_{\alpha_{j_k}}$ for $p = 1, \ldots, k$ we have $(F_1, \ldots, F_k) \subseteq \Gamma_{\alpha_{j_k}}$. Thus $\Gamma_{\alpha_{j_k}} \vdash B$ and $\Gamma_{\alpha_{j_k}} \not\vdash B$. This is a contradiction since $\Gamma_{\alpha_{j_k}}$ is consistent. Hence, $\Gamma'$ is consistent.

(Recall all $\Gamma_\alpha$ are consistent.)

If $\Gamma' \vdash \neg \forall B$, take $\Gamma_\beta = \Gamma'$. Otherwise take $\Gamma_\beta = \Gamma' \cup \{ \forall B \}$. $\Gamma_\beta$ is consistent by the previous theorem. Hence, by transfinite induction we have defined all the required $\Gamma_\alpha$ for $\alpha < \gamma$. Let $\overline{\Gamma} = \bigcup \{ \Gamma_\alpha \mid \alpha < \gamma \}$. By the previous argument $\overline{\Gamma}$ is consistent. Suppose there is a closed formula $C$ such that $C \not\in \overline{\Gamma}$ and $\overline{\Gamma} \cup \{ C \}$ is consistent. Recall
\( C = A_\mu \) for some \( \mu < \gamma \) since \( \gamma \) is the cardinality of the predicate calculus. Let \( \Gamma'' = \bigcup \{ \Gamma_a | a < \mu \} \).

Then \( \Gamma'' \) is consistent and \( \Gamma'' \subseteq \Gamma \). Hence, \( \Gamma'' \cup \{ C \} \) is consistent. Thus we can not have \( \Gamma'' \vdash \neg C \). Recall \( \neg C = \neg A_\mu \). For if \( \Gamma'' \vdash \neg A_\mu \) then \( \Gamma'', A_\mu \vdash \neg A_\mu \) and since \( \Gamma'', A_\mu \vdash A_\mu \), we would have \( \Gamma'' \cup \{ C \} \) inconsistent.

Since \( \Gamma'' \) does not derive \( \neg C \) and \( \Gamma'' = \bigcup \{ \Gamma_a | a < \mu < \gamma \} \), we would have \( \Gamma_C = \Gamma'' \cup \{ C \} \subseteq \Gamma \). So \( \Gamma \in \Gamma \). Thus \( \Gamma \) is a maximal consistent set of closed formulas. The proof is complete. Note: The union of consistent sets constructs \( \Gamma \) the maximal set.

Corollary. A set \( \Gamma \) of closed predicate formulas is consistent iff every finite non-empty subset \( \Gamma_0 \subseteq \Gamma \) is consistent.

Definition: A set \( \Gamma \) of closed formulas is called complete iff for any closed formula \( B \) either \( \Gamma \vdash B \) or \( \Gamma \vdash \neg B \).

Corollary: If \( \Gamma \) is a consistent set of closed formulas there is a maximal consistent set \( \Gamma \) of closed formulas such that \( \Gamma \subseteq \Gamma \) and \( \Gamma \) is complete.

From the construction we see that \( B \in \Gamma \) iff \( \Gamma \vdash B \) where \( B \) is a closed formula.
We now need the following definition.

Definition: Let PC and PC' be two predicate calculi. We say that PC' is an elementary extension of PC iff the variables $IV = IV'$, the predicates $P = P'$, the $n$-place functions $F_n = F'_n$ for $n > 0$ and the constants $F_0 \subseteq F'_0$. In this case we may consider every PF to be contained in PF' that is PC $\subseteq$ PF'.

Recall that our discussion refers to infinite sets. In other words we are dealing with infinite predicate calculi.

Let $L$ be a predicate calculus and suppose $L$ has cardinality $\gamma$. By a $\mu$ extension of $L$ where $\gamma \leq \mu$ we mean a predicate calculus $L'$ obtained from $L$ by adding a set of constants of cardinality $\mu$ to the constants of $L$. We assume that none of the new constants appear in $L$ in any manner. Note that the cardinality of $L'$ is $\mu$. We now need the following theorem.

Theorem: Let $\Gamma$ be a set of sentences (formulas that have no free variables) and suppose that $A$ has precisely one free variable $x$. If $b$ is a constant such that $b$ appears in no formula of $\Gamma$ and $\Gamma |- \forall x A$, then $\Gamma |- AxA$. 

75
PROOF: Let $D_1, \ldots, D_k$ be the derivation of $\exists^x A$ from $\Gamma$. Let $Y$ be a variable not in any one of the $D_1, \ldots, D_k$. Then $\exists^y D_1, \ldots, \exists^y D_k$ is a derivation. Thus we have $\Gamma \vdash \exists^y A$ and we extend the derivation by the step $\forall y \exists^y A$ and then use the fact that $\forall y \exists^y A \leftrightarrow \exists^x A$.

Let $\Gamma$ be a consistent set of sentences in $L$ and let $\forall y = \gamma$, then there is a set of sentences $\Gamma'$ in $L'$ such that $\Gamma'$ is consistent in $L'$, $\Gamma'$ is maximally consistent and complete and moreover if $\exists^x A$ is a sentence in $L'$ then $\exists^x A \in \Gamma'$ iff for each constant $b$ in $L'$, $\exists^x A \in \Gamma'$. A set with this property is called a Henkin set.

Lemma: $\Gamma$ is consistent in $L'$.

PROOF: Suppose we can find a formula $A$ such that $\Gamma \not\vdash L' A$ and $\Gamma \not\vdash L' \neg A$. Then there is a finite subset $\{B_1, \ldots, B_k\} \subseteq \Gamma$ such that $\{B_1, \ldots, B_k\} \not\vdash L' A$ and $\{B_1, \ldots, B_k\} \not\vdash L' \neg A$. As before let $D_1, \ldots, D_m$ be the derivation of $A$ and $E_1, \ldots, E_r$ be the derivation of $\neg A$. Then there exists at most a finite set $\{b_1, \ldots, b_p\}$ of constants of $L'$ not in $L$ which
appear in \( \{D_1, \ldots, D_m, E_1, \ldots, E_r\} \). Let \( Y_1, \ldots, Y_r \) be distinct variables not in \( \{D_1, \ldots, D_m, E_1, \ldots, E_r\} \) and hence not in \( A \). Let \( D'_1, \ldots, D'_m, E'_1, \ldots, E'_r, A' \) be the result of substitution \( y_j \) for \( b_j \) in the corresponding formulas \( (j=1, \ldots, p) \). Then as before \( D'_1, \ldots, D'_m \) is a derivation of \( A' \) and \( E'_1, \ldots, E'_r \) is a derivation of \( \neg A' \). But now these derivations are in \( L \) since the variables are the same in \( L \) and \( L' \). Hence, \( \Gamma \models_L A' \) and \( \Gamma \models_L \neg A' \) so we have a contradiction. \( \Gamma \) is supposed to be consistent.

We see that extending only the constants we can't create inconsistencies.

Let \( \{F_{\mu} | \mu < \gamma \} \) be a well-ordering of all the formulas of \( L' \) which have precisely one free variable. Note that if \( \mu < \gamma \) then \( \text{card}(\mu) < \text{card}(\gamma) = \gamma \). Assume the free variable of \( F_{\mu} \) is \( Y_{\mu} \). Choose the constants \( b_{\mu}, \mu < \gamma \) in the following way. Let \( b_0 \) be a constant not in \( F_0 \). Suppose we have found distinct constants \( b_{\nu}, \nu < \mu < \gamma \) such that for each \( \nu, b_{\nu} \) does not appear in any of the formulas \( F_{\tau}, \tau < \nu \). Observe that there are at most \( \text{card}(\mu) \) constants in \( \{F_{\nu} | \nu < \mu \} \) and at most \( \text{card}(\mu) b_{\nu} \) for \( \nu < \mu \). Hence, there is a constant \( b_{\mu} \) which is not in any of the \( \{F_{\nu} | \nu < \mu \} \). Nor is it equal to any
b_υ for υ ≤ υ. Thus by transfinite induction we have a set of distinct constants \( \{ b_υ | υ < γ \} \) such that for each \( υ < γ \) \( b_υ \) is not in any of the \( F_τ \) for \( τ < υ \). For each \( υ < γ \) let

\[
φ_υ = (¬∀_υ F_υ \Rightarrow \bigwedge b_υ).
\]

We are now prepared to do the Hassenjaeger-Henkin version of Gödel's completeness proof for the predicate calculus. First to build a Henkin set we do the following:

Define for each \( υ < γ \), \( Γ_υ = Γ'U(φ_υ | υ < v) \)

\[
Γ_0 = Γ'U(φ_0)
\]
\[
Γ_1 = Γ'U(φ_0, φ_1) \text{ etc.}
\]

We show that all the \( Γ_υ \), \( υ < γ \) are consistent in \( L' \). \( Γ_0 \) is consistent. Suppose \( Γ_0 \) is not consistent. Then for some formula \( A \), \( Γ_0 \models L' A \) and \( Γ_0 \models L' ¬A \). \( A = (¬A = ¬φ_0) \) is a tautology. Hence we have \( Γ_0 \models ¬φ_0 \), i.e., \( Γ, φ_0 \models L' ¬φ_0 \). Hence, \( Γ \models L' φ_0 = ¬φ_0 \) but \( (φ_0 = ¬φ_0) = ¬φ_0 \) is a tautology. Therefore, \( Γ \models L', ¬φ_0 \), i.e., if \( Γ_0 \) is inconsistent, \( Γ \) implies \( ¬φ_0 \). Completeness is for pure predicate calculus. We need the following tautologies:

\[
¬(¬A = ¬B) = ¬A
\]
\[
¬(¬A = ¬B) = B.
\]
Hence, $\neg \phi_0 = \neg \wedge \gamma_0 F_0$

Thus, $\Gamma \models_{\mathfrak{L}} \neg \wedge \gamma_0 F_0$

Hence $\Gamma \models_{\mathfrak{L}} \wedge \gamma_0 F_0$ but $\Gamma$ is consistent. This is a contradiction so $\Gamma_0$ is consistent.

Suppose for some $v < \gamma$ $\Gamma_v$ is inconsistent. Then there is a least $u$ such that $\Gamma_u$ is inconsistent. Notice that $u \neq 0$. For each $\rho < u$, $\Gamma_{\rho}$ is consistent. As before we can show that $\Gamma_u \models_{\mathfrak{L}} \neg \phi_u$. Let $D_1, \ldots, D_k$ be the derivation of $\neg \phi_u$ from $\Gamma_u$. Let $E_1, \ldots, E_r$ be the distinct formulas in $D_1, \ldots, D_k$ which come from $\Gamma_u$. For each $j$ there is a least $v_j$ such that $E_j \in \Gamma v_j$. At most one $v_j = u$. Let $\xi$ be the largest $v_j \neq u (\xi < u)$. Then $\Gamma_{\xi}, \phi_\mu \models_{\mathfrak{L}} \neg \phi_u$. The rest is as before. This leads to $\Gamma_{\xi}$ is inconsistent which is a contradiction since $\xi < u$. Hence, none of the $\Gamma_v$ are inconsistent.

Let $\Gamma = \bigcup \{ \Gamma_v \mid v < \gamma \}$ then $\Gamma$ is consistent. Let $\Gamma'$ be the completion of $\Gamma$. i.e., a maximal consistent set of sentences in $L'$ such that $\Gamma \subseteq \Gamma'$. $\Gamma'$ is a Henkin set. For suppose $F$ is a formula with precisely one free variable. Then $F = F_v$ for some $v$. If
Given a consistent set \( \Gamma \) of sentences in a predicate calculus \( L \) we can extend the calculus \( L \) to a calculus \( L' \) where the only difference is the addition of constants such that in \( L' \):

(a) \( \Gamma \) has a maximal complete consistent extension \( \Gamma' \) (\( \Gamma \subseteq \Gamma' \))

(b) \( \Gamma' \) is a Henkin set. i.e., if \( \forall x F \) is a sentence where \( F \) has \( x \) as a free variable then \( \forall x F \in \Gamma' \) iff for each constant \( b \) \( \forall b F \in \Gamma' \).

Theorem: Let \( \Gamma \) be a set of sentences in the predicate calculus \( L \). Then \( L \) is consistent iff \( \Gamma \) has a model.

80
(i.e., a structure with a valuation that makes each sentence in \( \Gamma \) true. If \( s \) is a valuation and \( a \in \Gamma \) then \( s(a) = 1 \)).

Observation: If \( \Gamma' \) has a model so does \( \Gamma \). So we must show any complete consistent Henkin set has a model.

Theorem: Lowenheim-Skolem Theorem:

Let \( \mathcal{L} \) be a predicate calculus and let \( \widehat{\Gamma} \subseteq \mathcal{L} \) be a complete, consistent maximal set of sentences which is a Henkin set, then \( \widehat{\Gamma} \) has a model. (We assume as always that \( \widehat{\Gamma} \) has constants.) Now to build the model.

PROOF: We construct a structure as follows:

Let \( A_0 \) be the set of constants of \( \mathcal{L} \). Let \( A_1 \) be the set of all closed terms in \( \mathcal{L} \). Let \( A \in \mathcal{L}^n \) and \( b_1, \ldots, b_n \in A_2 = A_0 \cup A_1 \). We define a function \( A^*: A_2^* A_2^* \cdots A_2^* \rightarrow (0,1) \) by \( A^*(b_1, \ldots, b_n) = 1 \) iff \( A(b_1, \ldots, b_n) \in \widehat{\Gamma} \). Notice that if \( t \) is a closed term then the value assigned to \( t \) is \( t \). Let \( B \) be a sentence of weight \( k \). We will show that \( B \in \widehat{\Gamma} \) iff each valuation \( s(B) = 1 \). We do it by induction on the weight of \( k \). Recall that weight is the total number of symbols \( \neg, \rightarrow, \land \) which appear in \( B \).
Case (i). If \( k = 1 \) and \( B = AxC \) where \( C \) is atomic, then \( C = Da_1a_2,\ldots,a_n \) where some \( a_i = x \) and the rest are terms. We may suppose without loss of generality that \( a_1 = x \). i.e., \( C = Dxa_2,\ldots,a_n \) where \( a_2,\ldots,a_n \) are closed terms. Then \( AxC \in \Gamma \) iff for each closed term \( b \) \( Dba_2,\ldots,a_n \in \Gamma \). (Henkin property.) Let \( AxC \in \Gamma \) and let \( s \) be any valuation. Then \( s(x) = d \) for some closed term \( d \). \( s(C) = 1 \) iff \( Dda_2,\ldots,a_n \in \Gamma \) hence \( s(C) = 1 \) and so \( s(AxC) = 1 \). Thus if \( AxC \in \Gamma \), \( s(AxC) = 1 \). Now suppose for every valuation \( s \), \( s(C) = 1 \), then for each closed term \( d \), \( Dda_2,\ldots,a_n \in \Gamma \) and then by the Henkin property \( AxC \in \Gamma \). Hence if \( s(AxC) = 1 \) for all \( s \) then \( AxC \in \Gamma \).

Case (ii). \( B = \neg C \). Since the weight of \( B \) is one, \( C \) contains no logical symbols. Since \( B \) is closed we may write \( C = Dt_1t_2,\ldots,t_n \) where \( t_1,\ldots,t_n \) are closed terms. Let \( s \) be a valuation, then \( s(Dt_1,\ldots,t_n) = 1 \) iff \( Dt_1,\ldots,t_n \in \Gamma \). If \( Dt_1,\ldots,t_n \in \Gamma \) then \( B \notin \Gamma \) or else \( \Gamma \) is inconsistent. Hence \( B \in \Gamma \) iff \( s(Dt_1,\ldots,t_n) = 0 \) iff \( s(B) = 1 \).

Case (iii). \( B = C \supset D \). Since \( B \) is a sentence so are \( C \) and \( D \). We see that for any valuation \( s \),
s(B) = 0 iff s(C) = 1 and s(D) = 0. C and D are atomic and this we have from the definition of the model. s(B) = 0 iff C ∈ \Gamma and D ∉ \Gamma. i.e., C ∈ \Gamma and \neg D ∈ \Gamma. But C ⇒ (\neg D ⇒ \neg (C \equiv D)) is a tautology and hence \neg (C \equiv D) ∈ \Gamma. Moreover it is clear that \neg (C \equiv D) ∈ \Gamma iff C ∈ \Gamma and \neg D ∈ \Gamma. \neg (C \equiv D) ⇒ C, \neg (C \equiv D) ⇒ \neg D are tautologies). B ∉ \Gamma iff C ∈ \Gamma and \neg D ∈ \Gamma. Hence, B ∈ \Gamma iff D ∈ \Gamma or C ∉ \Gamma. i.e., B ∈ \Gamma iff s(D) = 1 or s(C) = 0 i.e., B ∈ \Gamma iff s(B) = 1. Now assume there is a k > 1 such that for any sentence B, where for weight B < k we have B ∈ \Gamma iff s(B) = 1 for every valuation s. The arguments now are nearly the same as above.

A theorem can be derived from axioms. With this definition we now state a corollary to the previous theorem.

Corollary. Let L be a predicate calculus. Given a sentence A in L, A is a theorem iff for every structure and every valuation s in that structure, s(A) = 1.

PROOF: Suppose A has the property that s(A) = 1 for every model and every valuation s in that model.
If \( A \) is not a theorem, let \( \Gamma = \{ \neg A \} \). Then \( \Gamma \) is consistent. For if there is a formula \( B \) such that 
\[
\neg A \mid \neg B
\]
then \( \neg A \Rightarrow B \) and \( \neg A \Rightarrow \neg B \) are theorems and hence so are \( \neg B \Rightarrow A \) and \( B \Rightarrow A \).

But \( (\neg B \Rightarrow A) \Rightarrow [(B \Rightarrow A) \Rightarrow A] \) is a tautology. Hence \( A \) is a theorem.

If \( \Gamma \) is consistent it has a model, (i.e., a structure such that for each valuation \( s \) in the structure \( s(\neg A) = 1 \).) By assumption for each of these \( s \), \( s(A) = 1 \) and hence \( s(\neg A) = 0 \). This is a contradiction so any sentence \( A \) with the stated property is a theorem.
6. FLOWCHART PROGRAMS

We will now apply the preceding information about propositional calculi and predicate calculi to a study of program correctness. First the notion of a flowchart program is needed.

A predicate calculus with certain attributes is needed. We assume a predicate calculus, L, with integer arithmetic. The following are in L:

1) among the constants are the numerals, positive and negative.

2) among the functions are:
   i) $S$ where $S(x, y)$ means $x + y$
   ii) $M$ where $M(x, y)$ means $x * y$
   iii) $I$ where $I(x)$ means $-x$

3) among the predicates are:
   $<, >, \leq, \geq, =$

4) In $\Gamma$ we have the axioms for arithmetic. For example $\forall x \forall y \forall z (s(x, s(y, z)) = s(s(x, y), z))$ which is associativity for addition. We also assume:

$$(\forall y) ((\sum_{0}^{k} P) \land (\sum_{y}^{x} P \neq \sum_{y+1}^{x} P)) \neq \forall y (\sum_{y}^{x} P)$$

which is the axiom of mathematical induction.

85
Note: If we were working with reals we would need additional features such as:

The Archimedean Law - $\forall x \forall y ((0 < x) \land (0 < y) \Rightarrow \exists z \forall z (\text{Int}(z) \land (x < zy)))$.

Multiplicative inverse
Tell what an integer is by a predicate

To prevent us from being stuck in the rationals we would need to say $\forall x \forall y (x > 0 \Rightarrow y \cdot y = x)$.

There are other general statements we would need to make with respect to roots, Dedekind cuts, etc. As mentioned earlier we will assume integer arithmetic.

**Flow Chart Program** - admissible statements

We let $\overline{x}$, $\overline{y}$ stand for vectors of variables $\overline{x} = (x_1, \ldots, x_m)$, $\overline{y} = (y_1, \ldots, y_p)$, and $\overline{z} = (z_1, \ldots, z_q)$. Admissible statements have the following syntax.

```
n: start $\overline{y} + h(\overline{x})$
n: $\overline{y} + g(\overline{x}, \overline{y})$
n: Go To m
n: If $\phi(\overline{x}, \overline{y})$ then $n_1$ else $n_2$
n: $\overline{z} + k(\overline{x}, \overline{y})$ Halt
```

where $n$ is an integer; $\overline{x}$, $\overline{y}$, $\overline{z}$ are assumed to have a fixed number of components in any program;
h and g are assumed to be vectors of functions. For example \( h(\bar{x}, \bar{y}) = (h_1(x_1, \ldots, x_m, y_1, \ldots, y_p), \ldots, h_q(x_1, \ldots, x_m, y_1, \ldots, y_p)) \). Also \( h(\bar{x}, \bar{y}) \) indicates that all the free variables occur among \( \bar{x} \) and \( \bar{y} \). \( \psi(\bar{x}, \bar{y}) \) is a predicate formula with the same rules about \( \bar{x} \) and \( \bar{y} \). \( n_1 \) and \( n_2 \) are integers. \( \bar{x} \) will be the input and unchanged, \( \bar{y} \) will be the output, and the program works mostly with \( \bar{y} \) and \( \bar{z} \). \( n \) gives the states by means of statement numbers.

The approach used here is the Floyd, Hoare, Symanties which is a type of operational symantics done in the Inductive Assertion Method.

Substitution is viewed as concurrent substitution of all components. For example if \( \bar{x} = (x_1, \ldots, x_m) \) and \( \bar{y} = (y_1, \ldots, y_p) \) then \( \bar{y} + g(\bar{x}, \bar{y}) \) means \( \bar{u} + g(\bar{x}, \bar{y}) \) and then \( \bar{y} + \bar{u} \)

Here are the flowchart structures:

```
START
\[ \bar{y} + h(\bar{x}, \bar{y}) \]
\[ \text{T} \quad \varphi(\bar{x}, \bar{y}) \quad \text{F} \]
\[ \text{T} \quad \varphi(\bar{x}, \bar{y}) \quad \text{F} \]
To \( n_2 \)
To \( n_1 \)
```

```
\[ \bar{y} + g(\bar{x}, \bar{y}) \]
\[ \bar{z} + k(\bar{x}, \bar{y}) \]
HALT
```

87
Here is an example program:

\[ \bar{y} = (y_1, y_2), \bar{x} = (x_1, x_2), h(x_1, x_2) = (x_1, 0) \]

1. `start (y_1, y_2) + (x_1, 0)`
2. If \( y_1 > 0 \) then 5 else 3
3. \((y_1, y_2) + (y_1 + 1, y_2)\)
4. GOTO 2
5. \((z_1, z_2) + (y_1, y_2)\) Halt

The flowchart for this example is as follows:

```
START

(y_1, y_2) + (x_1, 0)

y_1 > 0

(z_1, z_2) + (y_1, y_2)

(y_1, y_2) + (y_1 + 1, y_2)

HALT
```

We can use as the output predicate \( \psi(\bar{x}, \bar{z}) \neq (x_1 = z_1) \). Now to explain the notion of program correctness. We start with a relationship concerning the input, an input predicate. Suppose \( \Phi(\bar{x}) \) is a predicate, then we say the program stops over \( \Phi \) for values of \( \bar{x} \) satisfying \( \Phi \) iff there is a
finite path starting at start through the graph of
the program such that for each program predicate $P$
on the path either $P$ or $\neg P$ is true for the current
values of $\overline{y}$ determining the path to take, and the
path ends at Halt.

The example given does not require a restrict-
tive predicate on the input. We could use for
example $\varphi(x_1, x_2) \neq (x_1 = -1, x_2)$ or $(x_1, x_2) \neq (x_1 - x_2 > 0)$ etc. This example really has any true
predicate as the input predicate. If the input
predicate is false for all possible elements, in
this case integers, then we have no input data.
Consider $\psi(\overline{x}, \overline{y}) \neq (x_1 = z_1)$ for the output predicate.
Others could be used such as $\psi(\overline{x}, \overline{z}) \neq [(x_1 = z_1^2) \land (z_1 \geq 0)]$.

The program starts with a value which makes the
input predicate true and ends with a value that makes
the output predicate true. Correct program means
that if it starts with a value which makes $\varphi$ true
it will halt with a value which makes $\psi$ true.

A program is said to be partially correct re-
lative to $\varphi$ and $\psi$ iff when given a value of $\overline{x}$ for
which $\varphi(\overline{x})$ is true, if the program halts, then
$\psi(\overline{x}, \overline{z})$ is true for the obtained value of $\overline{z}$. 
i.e. \( \Phi(x) \neq (\text{program halts} \land \Psi(x, z)) \) or \( \forall'(x) \land \text{program halts} \neq \Psi(x, z) \). This means that a program which never halts is partially correct if the input predicate is satisfied.

A program is totally correct relative to \( \Phi \) and \( \Psi \) iff \( \Phi(x) \neq \text{program halts} \land (\Psi(x, z)) \). Note, the correctness depends on the input predicate, the output predicate and halting.

The example given is correct if the input predicate is chosen so that it will halt and \( \Psi(x, z) \neq (x_1 = z_1) \) is satisfied. We can change the output predicate to obtain an incorrect program. Consider \( \Psi(x, z) \neq (x_1 = z_1, z_2 = 4) \).

We will now view the previous discussion in a more formal manner. Instructions will be of the following types:

1) \( n: \text{start} \overline{y} + f(x) \) Next
2) \( n: \overline{y} + g(x, \overline{y}) \) Next
3) \( n: \text{If } P(x, \overline{y}) \text{ then } n_1 \text{ else } n_2 \)
4) \( n: \overline{z} + h(x, \overline{y}) \) Halt
5) \( n: \text{GOTO } n_1 \)

where:

\( n \) is a statement label
Next represents the statement label of the next instruction in an instruction sequence. (The listing of the instructions).

A flowchart program is a finite sequence of instructions \( I_1, I_2, \ldots, I_m \) such that:

a) the statement labels are increasing

b) \( I_1 \) is a start instruction and the only one.

c) \( I_m \) is a Halt instruction and the only one.

d) We assume that in instructions of types 3 and 5, \( n_1 \) and \( n_2 \) are existing statement labels.

Let \( P = \{I_1, \ldots, I_m\} \) be a program. Let \( s \) be a model for the given \( r \) in the predicate calculus.

A program state in \( \mathcal{S} \) is a quadruple \((\bar{v}, \bar{s}, \bar{n}, \bar{f})\) where \( \bar{s} \) is an assignment of values for \( \bar{x} \), \( \bar{n} \) is an assignment of values for \( \bar{y} \), \( \bar{f} \) is an assignment of values for \( \bar{z} \) and \( \bar{v} \) is one of the statement labels.

A move is a pair \((v_1, \bar{s}(1), \bar{n}(1), \bar{f}(1))\)

\[(v_2, \bar{s}(2), \bar{n}(2), \bar{f}(2))\]

with the following properties:

- \( v_1 \) is the label of:
  - an instruction of Type 1, \( v_2 = \text{Next} (v_1) \)
  - \( \bar{n}(2) = f(\bar{s}(1)) \) where \( f \) is the interpreted function

91
corresponding to f. \( f^{(2)} \) is arbitrary.
\[ \bar{s}(2) = \bar{s}(1) \].

b) an instruction of Type 2, \( v_2 = \text{Next} \ (v_1) \),
\[ \bar{s}(2) = \bar{s}(1), \quad \bar{n}(2) = g(\bar{s}(1), \bar{n}(1)) \].

c) an instruction of Type 3, \( \bar{s}(2) = \bar{s}(1) \),
\[ \bar{n}(2) = \bar{n}(1), \quad \bar{f}(2) = \bar{f}(1), \quad \text{if } p(\bar{s}(1), \bar{n}(1)) = 0 \text{ then } v_2 = n_2 \text{ else } v_2 = n_1 \] where 0 is false.

d) an instruction of Type 4, \( v_2 = v_1, \quad \bar{f}(2) = h(\bar{s}(1), \bar{n}(1)), \quad \bar{s}(2) = \bar{s}(1), \quad \bar{n}(2) = \bar{n}(1) \).

c) \( v_2 = n_1 \) and all others remain the same for an instruction of Type 5.

An execution sequence consists of:

i) an assignment of \( \bar{s} \) for \( \bar{x} \), \( (\bar{s}(1) = \bar{s}) \)

ii) a sequence of quadruples \( q^1, q^2, \ldots, q^k \)
such that \( q^1 \) has the label of \( I_1 \), and has \( \bar{s} \) for its second component, \( q^i \) to \( q^{i+1} \) is a move, and \( q^k \) has the label of \( I_m \).

A generalized execution sequence is one which is finite or infinite without the requirement that the label of a Halt statement appear in it.

Let \( \xi(\bar{x}) \) be a well formed formula, then we say that \( P \) stops relative to \( \xi \) and \( \bar{s} \) iff.
1) $\mathcal{Q}^e(\overline{s}) = 1$ i.e. is true

2) The generalized execution sequence determined by $\overline{s}$ is an execution sequence.

3) $P$ stops relative to $\mathcal{Q}$ iff $P$ stops relative to $\phi$ for all $\overline{s}$ such that $\mathcal{Q}(\overline{s}) = 1$.

Given well formed formulas $\mathcal{Q}(\overline{x})$ and $\psi(\overline{x}, \overline{z})$ we say $P$ is partially correct relative to $\mathcal{Q}$ and $\psi$ iff for any $\overline{s}$ such that $\mathcal{Q}(\overline{s}) = 1$ if the generalized execution sequence determined by $\overline{s}$ is an execution sequence (i.e. it stops) then $\psi(\overline{s}(1), \overline{f}(k)) = 1$ i.e. if $P$ stops relative to $\mathcal{Q}$ then $\psi(\overline{s}(1), \overline{f}(k)) = 1$.

$P$ is totally correct iff $P$ stops relative to $\phi$ and $\psi(\overline{s}(1), \overline{f}(k)) = 1$.

In order to establish the partial correctness of a program we need to find a predicate $\phi$ which remains invariant throughout the program with respect to $\overline{x}$, $\overline{y}$, and $\overline{z}$ and satisfies $\psi$ and is satisfied by $\mathcal{Q}$. Recall that $\psi$ is the output predicate and $\mathcal{Q}$ is the input predicate. In order to clarify the statement made above consider the following examples.
The input predicate is \( \mathcal{P}(x_1, x_2) \not\equiv (x_1 \geq 0) \land (x_2 \geq 0) \)

The output predicate is \( \mathcal{V}(x_1, x_2, z_1, z_2) \not\equiv x_1 = 
\begin{align*}
  (z_2 \times x_2 + z_1) & \land (0 \leq z_1) \land (z_1 < x_2)
\end{align*} \)

It turns out that \( z_1 = x_1 \mod x_2 \) and \( z_2 = x_1 \div x_2 \).

In an attempt to determine \( a \) we make cuts at appropriate positions of the flowchart program and work backwards as follows.
At cut C in the diagram we put \((x_1, x_2, y_1, y_2) \neq (x_1 = y_2 \cdot x_2 + y_1) \land (0 \leq y_1) \land (y_1 < x_2)\) because this must be the case at that point. Backing up to position B we put \(\psi(x_1, x_2, y_1 + x_2, y_2 - 1) \neq (x_1 = (y_2 - 1) \cdot x_2 + y_1 + x_2) \land (0 \leq y_1 + x_2)\) which must be the case at this point if \(\psi\) is to be satisfied at the end. Notice that this reduces to \((x_1 = y_2 \cdot x_2 + y_1) \land (-x_2 \leq y_1) \land (y_1 < 0)\). Moving back another step to cut A we see that \(\alpha\) must be a loop invariant such that:

\[
\alpha(x_1, x_2, y_1, y_2) \land (y_1 \geq 0) \neq \alpha(x_1, x_2, y_1 - x_2, y_2 + 1) \land (y_1 < 0) \neq \psi(x_1, x_2, y_1 + x_2, y_2 - 1).\]

Now if we can determine such an \(\alpha\) we will have proven the program to be partially correct. After studying the situation we try \(\alpha(x_1, x_2, y_1, y_2) \neq (x_1 = y_2 \cdot x_2 + y_1)\). Notice that if \(\phi\) is satisfied then \(\alpha\) is satisfied at the start since \(\alpha(x_1, x_2, x_1, 0) \neq x_1 = 0 \cdot x_2 + x_1\). We now test \(\alpha\) through the true portion of the loop. This means \(\alpha(x_1, x_2, y_1 - x_2, y_2 + 1) \neq (x_1 = (y_2 + 1) \cdot x_2 + y_1 - x_2) \neq (x_1 = y_2 \cdot x_2 + y_1)\) so \(\alpha\) passes the test here. Now out of the false trial. We need \(\alpha(x_1, x_2, y_1, y_2) \neq (x_1 = y_2 \cdot x_2 + y_1) \land (y_1 < 0) \land (-x_2 \leq y_1)\) because we found this was the
situation that must exist at point B. « does not give us \((-x_2 \leq y_1)\) or equivalently \((y_1 + x_2 \geq 0)\) so we see that we must add this condition to «. We must now go back and test through the program again in the same manner with:

\[\sigma(x_1, x_2, y_1, y_2) \neq (x_1 = y_2 \cdot x_2 + y_1) \land (y_1 + x_2 \geq 0)\]

In doing so we see that the previous tests still hold and now at cut B we have \((x_1 = y_2 \cdot x_2 + y_1) \land (y_1 < 0) \land (y_1 + x_2 \geq 0) \neq (x_1 = y_2 \cdot x_2 + y_1) \land (y_1 < 0)\) or in other words \(\sigma(x_1, x_2, y_1, y_2)\) now give us the necessary conditions at B. It follows that « holds at C with the necessary conditions satisfied because \((y_1 \geq 0)\) since \(y_1 + x_2\) replaces \(y_1\) and \(y_1 + x_2 \geq 0\), also \((y_1 < x_2)\) since when \(y_1 + x_2\) replaces \(y_1\) the \(y_1\) in \(y_1 + x_2\) is negative, and also \(x_1 = y_2 \cdot x_2 + y_1\) since \((y_2 - 1) \cdot x_2 + y_1 + x_2 = y_2 \cdot x_2 + y_1\).

Having found « we have established that the program is partially correct. It is not totally correct because if \(x_2 = 0\) the program will never halt. If we changed the input predicate to:

\(\varphi(x_1, x_2) \neq (x_1 \geq 0) \land (x_2 > 0)\) the program would be totally correct.
Here is another example.

\[ (y_1, y_2, y_3) + (0, 0, 1) \]

\[ y_2 + y_2 + y_3 \]

\[ y_2 > x \]

\[ (y_1, y_2, y_3) + (y_1 + 1, y_2, y_3 + 2) \]

\[ x = x_1 \]

\[ y = (y_1, y_2, y_3) \]

\[ z = z \]

\[ \Phi(x) = x \geq 0, \quad \Psi(x, z) = (z^2 \leq x < (z + 1)^2) \]

At first it is difficult to see what this program is doing. If we consider the series 1 + 3 + 5 + ... + (2n + 1) = (n + 1)^2 we see how the program works. Notice that \( y_1 = n \) and \( y_2 = 1 + 3 + ... + (2n + 1) = (n + 1)^2 \). Another way to express the
result is \( z = \lfloor \sqrt{x} \rfloor \).

Backing up through the program we see that at

\( D \) on the flowchart we need \( (y_1^2 \leq x) \land (x < (y_1 + 1)^2) \).

We will try this for \( \preceq \). That is \( \preceq (x, \bar{y}) \equiv (y_1^2 \leq x) \land (y_2 = (y_1 + 1)^2) \). Notice that if we go to cut \( B \) the condition holds the first time through since

\( \preceq (x, y_1, y_2 + y_3, y_3) \equiv (y_1^2 \leq x) \land ((y_2 + y_3) = (y_1 + 1)^2) \) and with \( (x, 0, 1, 1) \) this says \( (0^2 \leq x) \land (1 = (0 + 1)^2) \) which is true.

Now if we go through the false path of the loop and return to \( B \) does \( \preceq \) still hold? In other words if \( (x, y_1, y_2, y_3) \equiv (x, y_1 + 1, y_2 + y_3 + 2, y_3 + 2) \) and \( y_2 \leq x \) and \( (y_1^2 \leq x) \land (y_2 = (y_1 + 1)^2) \) does \( ((y_1 + 1)^2 \leq x) \land (y_2 + y_3 + 2 = (y_1 + 2)^2) \) hold?

The answer is yes if \( y_3 = 2y_1 + 1 \). This follows from \( y_2 = (y_1 + 1)^2 \leq x \) and a little algebra on the second part. Hence we must add to \( \preceq \) that

\( y_3 = 2y_1 + 1 \). Checking back with the new \( \preceq (x, y_1, y_2, y_3) \equiv (y_1^2 \leq x) \land (y_2 = (y_1 + 1)^2) \land (y_3 = 2y_1 + 1) \) we see that it holds at \( B \) the first time through since \( 1 = 2(0) + 1 \). Therefore \( \preceq \) is a loop invariant for this loop and we now need to see if the proper conditions for \( y \) are met when we exit the loop.
If \( (x, \bar{y}) \land (y_2 > x) \) then \( (y_1^2 \leq x) \land (x < (y_1 + 1)^2) \) since \( y_2 = (y_1 + 1)^2 \) by \( (x, \bar{v}) \). This is what we needed at D so the program is partially correct.

At this point we need to describe the process of finding the cuts in a more formal manner. Introduce cut points as follows:

a) A cut point immediately after start.

b) A cut point immediately prior to the last assignment before Halt.

c) In each loop at least one cut point to disconnect the loop. In most cases this cut is just prior to a test.

In a sense we are unfolding or unrolling the loops out into infinite paths. The arcs of the flowchart are those directed paths connecting cut points but having no intermediate cut points. We designate these arcs by \( a, b \) and \( a', b' \) etc. At point \( a \) we have conditions on the variables so that we can traverse the arc. That is, we specify certain conditions at \( b \) and try to find conditions at \( a \) which will allow the traverse. We work backwards.

Given the arc \( s \), joining \( a \) and \( b \) we construct
at each statement two objects, \( R(\overline{x}, \overline{y}) \) a WFF having \( \overline{x} \) and \( \overline{y} \) as the only free variables and a term \( r(\overline{x}, \overline{y}) \) where \( \text{dim}[r(\overline{x}, \overline{y})] = \text{dim} \overline{y} \).

The construction is made as follows relative to each of the following cases:

\[
\begin{align*}
&\downarrow \\
&\overline{y} + g(\overline{x}) \\
&\downarrow \\
&\overline{y} + h(\overline{x}, \overline{y}) \\
&\downarrow \\
&T \\
&\text{P}(\overline{x}, \overline{y}) \\
&\downarrow \\
&R(\overline{x}, \overline{y}), \text{r}(\overline{x}, \overline{y}) \\
&\downarrow \\
&\text{P}(\overline{x}, \overline{y}) \\
&\downarrow \\
&R(\overline{x}, \overline{y}), \text{r}(\overline{x}, \overline{y}) \\
&\downarrow \\
&\neg \text{P}(\overline{x}, \overline{y}) \lor \text{R}(\overline{x}, \overline{y}), \text{r}(\overline{x}, \overline{y}) \\
&\downarrow \\
&R(\overline{x}, \overline{y}), \text{r}(\overline{x}, \overline{y})
\end{align*}
\]

If the conditions given prevail before entering the rectangle then the conditions given after the
Given an arc from a to b, start at b with $R(\bar{x}, \bar{y}) \neq \text{True}$ and $r(\bar{x}, \bar{y}) = \bar{y}$. This process will determine at a the pair $R_s(\bar{x}, \bar{y})$ and $r_s(\bar{x}, \bar{y})$. The assignments change the variable states and the test conditions change the predicate.

At each node a we assign a predicate $I_a(\bar{x}, \bar{y})$ where $I_a(\bar{x}, \bar{y}) = \phi(\bar{x})$ the input WFF. Also $I_b(\bar{x}, \bar{y}) = \psi(\bar{x}, h(\bar{x}, \bar{y}))$ where the final statement is $\bar{z} + h(\bar{x}, \bar{y})$ Halt. If we can establish on each arc $s$ connecting a to b that in the given model

$$\Lambda x \Lambda y [R_s(\bar{x}, \bar{y}) \land I_a(\bar{x}, \bar{y}) \neq I_b(\bar{x}, r_s(\bar{x}, \bar{y}))],$$

note: $\Lambda \bar{x}$ means $\Lambda x_1, \Lambda x_2, \ldots \Lambda x_p$, where if $s$ connects a to b we require $\Lambda \bar{x} \Lambda \bar{y} [R_s(\bar{x}, \bar{y}) \land \phi(x) \neq I_b(\bar{x}, r_s(\bar{x}, \bar{y}))$ and if $s$ connects a to b we require $\Lambda \bar{x} \Lambda \bar{y} [R_s(\bar{x}, \bar{y}) \land I_a(\bar{x}, \bar{y}) \neq \psi(\bar{x}, r_s(\bar{x}, \bar{y}))$ then the program is partially correct.

Consider the following example:
Formally odd(x) means \((\forall y)(x = 2y + 1)\) end \(y_1y_3\) means \(y_1 \times y_3\). This program calculates \(x_1^{x_2}\), that is \(x_1\) to the \(x_2\) power.
Here is a PASCAL version of this program.

```pascal
y3 := 1;
while y2 > 0 do begin
  if odd(y2) then begin
    y3 := y3*y1;
y2 := y2 - 1
  end else begin
    y1 := SQ(y1);
y2 := y2 DIV 2
  end
end
end
```

Notice the arcs that the flowchart program has been divided into. They are \( s_0 \) from \( A \) to \( a \), \( s_1 \) from \( a \) back to \( a \) through the T path, \( s_2 \) from \( a \) back to \( a \) through the F path, and \( s_3 \) from \( a \) to \( B \).

Working backward the inductive assertions lead us to try \( I_a(x, y) \Rightarrow (x_2 \geq 0) \land (y_2 \geq 0) \land (y_3 \cdot y_1 \cdot y_2 = x_1 \cdot x_2) \). We now trace through the arcs.

At \( a \) we have \( R \) : true

\[ r = (y_1, y_2, y_3) \]

and at \( A \)

\[ R_{s_0} \] : true

\[ r_{s_0} = (x_1, x_2, 1) \]
Now do we have:
\[ \lambda \bar{x} \lambda \bar{y} (P(x) \pm I_a(\bar{x}, r_{s_0}(\bar{x}, \bar{y}))) \text{ or } \lambda \bar{x} \lambda \bar{y} (x_2 > 0 \pm (x_2 > 0) \land (x_2 > 0) \land (1 \times x_1 x_2 = x_1 x_2))? \] This is true so \( s_0 \) is taken care of.

Now for arc \( s_1 \):

We ask does:
\[ \lambda \bar{x} \lambda \bar{y} (\text{odd}(y_2) \land (y_2 \neq 0) \land I_a(\bar{x}, \bar{y}) \neq (x_2 \geq 0) \land (y_2 - 1 \geq 0) \land (y_1 \times y_3 \neq y_1 x_1 x_2))? \] Checking this we see that it is true so we have verified the case for \( s_1 \).
Now for arc $s_2$

\[ R_{s_2} : (y_2 \neq 0) \land \neg \text{odd } (y_2) \]

\[ r_{s_2} : (y_1, y_1, y_2/2, y_3) \]

\[ R : \neg \text{odd } (y_2) \]

\[ r : (y_1, y_1, y_2/2, y_3) \]

\[ R : \text{True} \]

\[ r : (y_1, y_1, y_2/2, y_3) \]

\[ (y_1, y_2, y_3) \lor (y_1, y_1, y_2/2, y_3) \]

\[ R : \text{True} \]

\[ r : (y_1, y_1, y_2, y_3) \]

We ask does:

\[ \forall x \forall y ((y_2 \neq 0) \land \neg \text{odd } (y_2) \land I_d (x, y) \land I_u (x, y) = I_u (x, \bar{y})) \]

or does \[(y_2 \neq 0) \land \neg \text{odd } (y_2) \land (x_2 \geq 0) \land (y_2 \geq 0) \land (y_3 \cdot y_1 = x_1 \cdot x_2) \land (x_2 \geq 0) \land (y_2/2 \geq 0) \land (y_3 \cdot (y_1 \cdot y_1) y_2/2) = x_1 \cdot x_2)? \]

Checking this we see that it is true so we have verified the case for $s_2$. 

105
Now for $s_3$:

$$R_{s_3} : (y_2 = 0)$$
$$r_{s_3} : (y_1, y_2, x_1^{x_2}) \text{ or }$$
$$\quad (h(\bar{x}, \bar{y})) = (y_3)$$

$$T$$

$$z + y_3$$

$$\nabla(\bar{x}, \bar{z}) \pm z = x_1^{x_2}$$

Notice that here we need to look at $h$ as a function which projects the third component of $\bar{y}$. We ask does $(y_2 = 0) \land I_a(\bar{x}, \bar{y}) \neq \nabla(\bar{x}, r_{s_3} (\bar{x}, \bar{y})) \text{ or }$ $(y_2 = 0) \land (x_2 \geq 0) \land (y_2 \geq 0) \land (y_3 \cdot y_1^{x_2} = x_1^{x_2}) \neq$ $\nabla(\bar{x}, y_3) \neq y_3 = x_1^{x_2}$? The answer is obviously yes so we are done.

It follows that the program is partially correct and since $y_2$ must eventually equal zero the program is totally correct.
Without having specified it, we have been using the notion of a well founded set. A well founded set is a set with a P.O. relation such that there is no infinite descending sequences contained in it. The positive or non-negative integers are examples of well founded sets.
7. CONCLUSIONS

We find that the predicate calculus needed for the study of program correctness must have certain attributes. We have assumed a predicate calculus, $L$, with integer arithmetic. This could be extended to include real arithmetic if some additional proportions were included.

It is necessary to carefully define program correctness. We start with an input predicate over the input variables, $\varphi(X)$. With this input predicate we associate an output predicate over the input variables and the output variables, $\psi(X, Z)$. A program is said to be partially correct relative to $\varphi$ and $\psi$ iff when given a value of $X$ for which $\varphi(X)$ is true, if the program halts, then $\psi(X, Z)$ is true for the obtained value of $Z$. If this condition is met and we can prove the program will halt, it is totally correct. We find that the methods used to prove a program partially correct do not prove the program will halt. It follows that the halting problem is separate.

In order to establish the partial correctness of a program we need to find a predicate $\xi$ which remains invariant at certain critical points throughout the
program. We do not have a formal theory for determining the cut points or $\alpha$. Some general rules can be followed which are helpful in determining cut points and $\alpha$ can be approached by tracing backward through the flowchart program starting with $\psi(X, Z)$. 
8. LIST OF REFERENCES


9. VITA

The author was born to Mr. and Mrs. Walter K. Krauss on February 2, 1944 in Woodbury, New Jersey. He earned his B.A. degree in History from Oklahoma College of Liberal Arts in 1967 and his M.A. in Mathematics from the University of Oklahoma in 1970. In the Fall of 1980, he began graduate study in Computing Science at Lehigh University. He has been teaching as a full-time faculty member in Mathematics, and now Data Processing, at Northampton County Area Community College in Bethlehem since 1970. He was promoted to full Professor in March 1982.