Math 260ABC
Introduction to Mathematical Logic
Fall 1988-Sprint 1989
Instructor: Sam Buss

Department of Mathematics
University of California, San Diego

Lecture Notes written by

Bruce Wieand

Computation

Math 260A - Mathematical Logic

September 23, 1988

This course will deal with the notions of truth, proof, and computation. It will combine ideas from the philosophy of mathematics, mathematics itself, and computer science.

"Effective computation" is a philosophical notion. The following are characteristics of effective computation:

- there is an effective process for finding an answer,
- it proceeds in a deterministic fashion; i.e.,
    - no luck or randomness
    - no appeal to deities
    - no intuition
- a finite amount of information fully describes procedure;
- however time and space considerations may make it impractical.

Feasible computation is "practical" effective computation. As an example, factoring an integer is effectively computable; whether or not it is feasible is an open question.

A function mapping $N \to N$ might be effectively computable without having any proof that it is effectively computable. For example, let $h : N \to N$ be a function which is not effectively computable, and let

$$f(x) = \begin{cases} 0 & \text{if Riemann's hypothesis is true} \\ h(x) & \text{otherwise} \end{cases}$$

Then f might be effectively computable, but we can't prove it (yet). On the other hand,

$$g(x) = \begin{cases} 0 & \text{if Riemann's hypothesis is true} \\ 1 & \text{otherwise} \end{cases}$$

is effectively computable because it is a constant function (but we don't know which constant it is, yet). Finally, consider the decimal expansion of π. There is an effective procedure for writing out the decimal expansion of π, but it never halts. If instead we let $f(n) = n^{\text{th}}$ digit of ϕ, then f is effectively computable. These two notions are different but essentially equivalent since we could use either one to accomplish the other.

Definition: A function $f : N \to N$ is *effectively computable* iff there is an effective procedure which for every $n \in N$ will produce (i.e., eventually halt) the value $f(n)$.

Definition: A *relation* is a subset of N^k. The *characteristic function* of a relation R is

$$\chi_R(\vec{n}) = \begin{cases} 1 & \text{if } \vec{n} \in R \\ 0 & \text{otherwise} \end{cases}$$

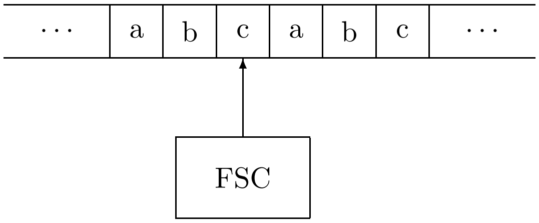
A relation is *decidable* or *solvable* iff its characteristic function is effectively computable.

Turing Machines

Introduced in 1936 by Alan Turing, Turing machines (TM) are characterized by

- a finite state control,
- an infinite tape with cells that can hold one symbol from a finite alphabet, and
- a read/write head which can read and write symbols and can move left or right one cell at a time.

Pictorially, we have



Formally, a TM consists of

- a finite set Σ of cardinality ≥ 2 called the alphabet,
- a finite set Q of states with $Q \cap \Sigma = \emptyset$,
- the symbols L and R with $L, R \notin Q \cup \Sigma$, and
- a partial function $f : Q \times \Sigma \to (\Sigma \cup \{L, R\}) \times Q$.

So $f(q_i, a_j) =$ (a new symbol in Σ or L or R, a new state). By convention, Σ contains the blank symbol, $\flat$. Also, by convention, when a TM starts, all but a finite number of cells on the tape contain $\flat$'s.

An instantaneous description (ID) of a TM computation is

- an infinite 2-way tape with a symbol in each square,
- a current state, q_i, and
- a current tape head position with the square under the tape head containing the symbol a_i.

At the next instant of time, the next ID is an action and a new state, defined by $f(q_i, a_j)$ according to the following rules.

- If action $\in \Sigma$, then a_j is changed to this symbol, and the tape head remains at the same position.
- If action is L or R, then a_j is unchanged and the tape head moves left or right one square.
- The next state is the new state.
- If $f(q_i, a_j)$ is undefined, then the TM is in a halting configuration.

In comparing a TM with a modern day computer, we find that modern day computers have more powerful instructions, they have random access memory, and they can access more than one memory location per instruction. But TM's have an infinite amount of memory.

Turing Computability

Math 260A - Mathematical Logic

September 26, 1988

As a simple example of a TM, consider a TM which operates on a string of 1's. The alphabet, $\sigma = \{\flat, 1\}$. We want the TM to halt over the rightmost 1 in the string. If the TM starts on a $\flat$, then it halts without doing anything. The transition function can be defined as follows:

$q_0 1 : Rq_0$
$q_0 \flat : Lq_1$

with $f(q_1, \flat) = f(q_1, 1) =$ undefined. To handle the last condition, we need

$q_{-1} 1 : Rq_0$

and $f(q_{-1}, \flat) =$ undefined. So $Q = \{q_{-1}, q_0, q_1\}$ with q_{-1} the initial state. Another method for defining a TM can be found in an example in Boolos & Jeffrey on page 24; there, they use a transition diagram for defining a doubling function.

Definition: The numbers (in unary) $x_1, \ldots, x_n > 0$ are input in *standard notation* if the TM starts with x_1 1's, $\flat$, x_2 1's, $\flat$, …, $\flat$, x_n 1's, and otherwise all $\flat$'s, and with the tape head at the leftmost 1.

So, for example, the string $\ldots\flat 111 \flat 11 \flat \ldots$ codes the pair $(3, 2)$.

Definition: A function $f : (N^+)^k \to N^+$ ($N^+ = \{1, 2, \ldots\}$) is *turing computable* iff there is a TM M such that for all $x_1, \ldots, x_n \in N^+$ if M is started with $x_1, \ldots, x_n$ in standard notation, then M eventually halts with $f(x_1, \ldots, x_n)$ output in standard notation.

Example: $n \mapsto 2n$ is turing computable by the example on page 24 of Boolos & Jeffrey.

Example: The TM which just halts computes the identity $n \mapsto n$. On two or more inputs, its output is not in standard notation.

Definition: A relation $R \subseteq (N^+)^k$ is turing computable or *turing decidable* iff

$$\chi_R(\vec{x}) = \begin{cases} 1 & \text{if } \vec{x} \in R \\ 0 & \text{otherwise} \end{cases}$$

is turing computable.

Church's Thesis: Turing computable $\equiv$ effectively computable. That turing computable $\Rightarrow$ effectively computable seems clear, but the converse is not so obvious. But so far, *every* notion of effectively computable has turned out to be turing computable. Remember, feasible computation is a much stronger notion then effectively computable.

Example: of a more complicated TM. Find a TM which starts on the leftmost symbol of a string of 0's and 1's, and "palindromizes" it by appending a reversed copy of it to itself. We'll use a larger alphabet, $\Sigma = \{\flat, 0, 1, 0^{'}, 1^{'}\}$ to keep track of where we are in the copying. To get started, we'll first design an algorithm:

1. Scan right to the first $\flat$.
2. Go left to the first unprimed symbol. If none, go to step 7.
3. Prime the symbol and remember it.
4. Go right to the first $\flat$.
5. Put the primed symbol there.
6. Go to step 2.
7. Unprime everything, go to the leftmost 0 or 1, and halt.

On input 01011, each loop of this algoritm would append one symbol:

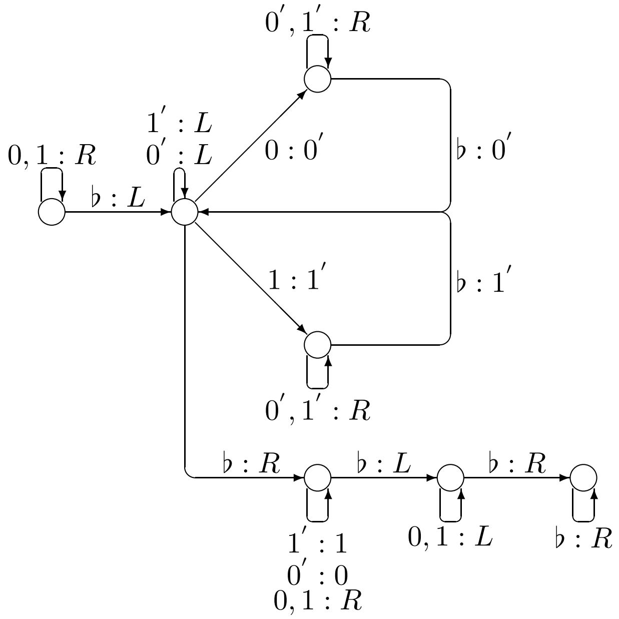
01011
$01011^{'}1^{'}$
$0101^{'}1^{'}1^{'}1^{'}$

$\vdots$

$0'1'0'1'1'1'1'0'1'0'$

0101111010

With the algorithm, we can then design the TM:



Turing Computability, Uncomputability

Math 260A - Mathematical Logic

September 28, 1988

Last time we saw an example of a TM that used the extra symbols $0'$ and $1'$. We didn't have to use extra symbols. The next theorem shows that turing machines, as models of computation, are insensitive to the number of alphabet symbols (as long as there are at least two symbols). Turing machines are also insensitive to other details of the definition. Some other variants are

- more than one tape,
- more than one tape head,
- a tape with more than one dimension, and
- a random access tape.

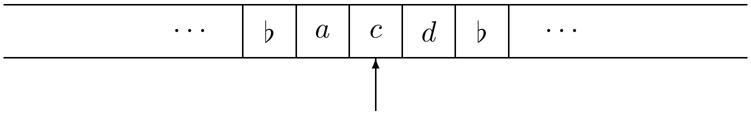
Theorem: Suppose that f is turing computable. Then f is computed by some TM which has $\Sigma = \{\flat, 1\}$.

Proof: The idea is to code the symbols of the larger alphabet with a set of $\flat$'s and 1's. Suppose that f is computed by M_0 with Σ_0 of size s and states $q_1, \ldots, q_t$. Then we want to build a TM M which also computes f and has $\Sigma = \{\flat, 1\}$. A symbol from Σ_0 will be coded by $\lceil \log_2 s \rceil$ $\flat$'s and 1's - a fixed length binary code.

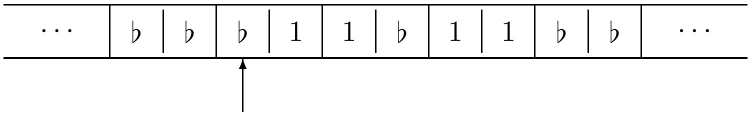
For example, suppose that $\Sigma_0 = \{\flat, a, c, d\}$. Then $s = 4$, and $\log_2 s = 2$. So the codes are:

$\flat$	-	$\flat\flat$
a	-	$\flat 1$
c	-	$1\flat$
d	-	11

If M_0's tape is



then M's tape is

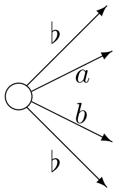


M has t states, $q'_1, \ldots, q'_t$, corresponding to $q_1, \ldots, q_t$. Whenever M is in state q'_j, M's tape head is at the leftmost bit of a code. In state q'_j, M moves right $k-1$ (let $k = \lceil \log_2 s \rceil$) squares remembering the contents of them. It ends up in one of s states unique to q_j. This gives us

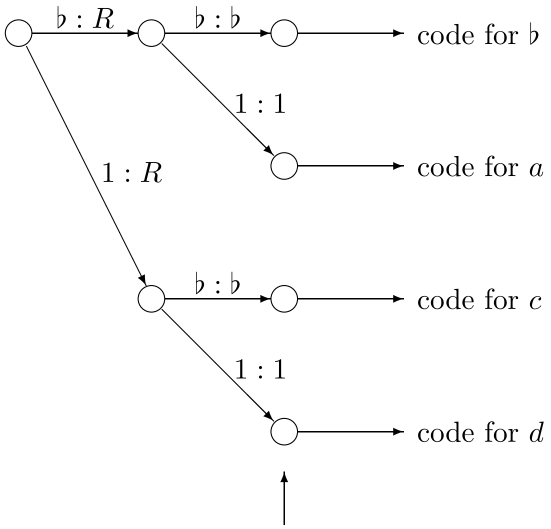
t	states $q'_1, \ldots, q'_t$,
$2t$	states after moving one square,
$4t$	states after moving two squares,
$\vdots$	
st	states after moving $k-1$ squares,

for a total of $(2^k + s - 1)t$ states so far.

For example, if in M_0 we had



then in M we would have



s states for each $q_i^{'}$
(st states for $q_1^{'}, \ldots, q_t^{'}$)

Each of the st states corresponds to M_0 reading a symbol from Σ_0 in a state q_j. M_0 would now do either

1. move left or right, or
2. overwrite the current symbol,

and possibly change state. So M has to either

1. move left $2k-1$ squares or move right one square, or
2. move left $k-1$ squares overwriting the code as it goes,

and possibly change to a new state. This gives

$$((2k-1)+1+s(k-1))t = (2k+s(k-1))t$$

more states. Without loss of generality, assume that $\flat \in \Sigma_0$ is coded by k $\flat$'s and that $1 \in \Sigma_0$ is coded by k 1's. Then M does the following.

- Stretch its input by changing $\flat$'s and 1's to k $\flat$'s and k 1's.[1]
- Simulate M_0 as above until a halting configuratoin is reached.
- "Smash" the output back down. □

Many functions are Turing computable. $f(a,b) = a+b$ for $a,b > 0$ is effectively computable; hence, by Church's thesis, it is Turing computable. So is $g(n) = 2^n$. And so is $h(n) = 2 \Uparrow n$, where

$$x \Uparrow y = \underbrace{x^{x^{\cdot^{\cdot^{\cdot x}}}}}_{y\ x'\mathrm{s}}.$$

(Although h is very infeasible.) So a question to ask is "is there an integer function which is not Turing computable?" The answer is yes, and now we'll find one.[2]

Definition: A TM M has *productivity* n if, when started in its initial state with a completely blank tape, it eventually halts with n output in standard notation. Otherwise, M's productivity is 0.

Definition: The function $BB : N^+ \to N^+$ is defined by $BB(n) =$ the maximum productivity of any n state TM with alphabet $\{\flat, 1\}$. (BB stands for busy beaver.)

Claim: There are only finitely many such TM's for any fixed n. So $BB(n)$ = the maximum of a finite set.[3] And so $BB(n)$ is well defined.

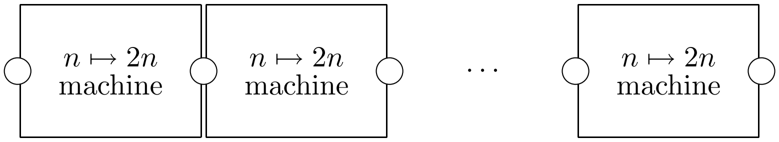
Claim: BB is not Turing computable, and hence, by Church's thesis, not effectively computable. (We'll prove this next time.)

The problem with BB is that it grows too fast to be computable. As a lower bound on $BB(n)$, recall that we have a 12 state TM which computes $n \mapsto 2n$. Now, if we concatenate k copies of this machine;

[1] It shouldn't try to double the $\flat$'s on either end of the input; otherwise, it would loop forever. Remember, Turing computable means that the input consists of strings of 1's separated by $\flat$'s (standard input).

[2] Note: without Church's thesis, it is probably impossible to give a concrete example of a function which is provably not effectively computable.

[3] This is the reason for limiting TM's to finite alphabets; i.e., so we don't have to find the maximum of an infinite set.



This machine has $11k+1$ states, and outputs 2^k 1's in standard notation. So

$$\begin{aligned} BB(11k+1) &\geq 2^k, \text{ or} \\ BB(n) &\geq 2^{\lfloor \frac{n-1}{11} \rfloor} \\ &\geq 2^{\frac{n}{11}-1}. \end{aligned}$$

Uncomputability, The Halting Problem

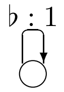
Math 260A - Mathematical Logic

September 30, 1988

Last time, we introduced the busy beaver function $BB(n)$ = the largest number (in standard notation) output by some n-state TM started on a blank tape with alphabet $\{\flat, 1\}$. We then constructed a TM with $11k+1$ states that output 2^k. This gave us the lower bound $BB(11k+1) \geq 2^k$. And so, $BB(n) \geq 2^{\lfloor \frac{n-1}{11} \rfloor} \geq 2^{\frac{n}{11}-1}$ for all n.

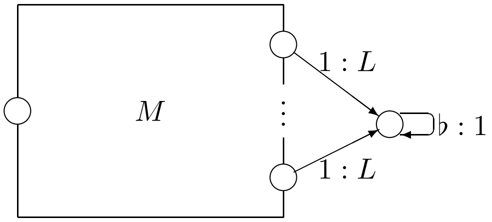
Fact: $BB(n+1) \geq BB(n)$ for all n. This is obvious since an n-state TM which outputs $BB(n)$ can have an unreachable state added to it.

Theorem: $BB(1) \geq 1$.
Proof:

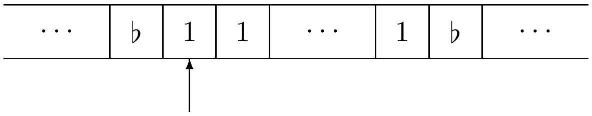


□

Theorem: For all n, $BB(n+1) > BB(n)$.
Proof: Suppose that M is an n-state TM machine with productivity $BB(n)$. Then modify M to get M^* with $n+1$ states of productivity $BB(n)+1$ as follows:



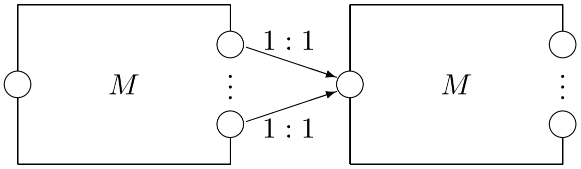
So, for every halting state of M, we add transitions which move left one square and write a 1. So when M halts in the configuration



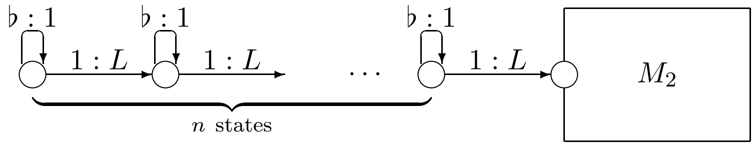
with $BB(n)$ 1's, M^* moves the tape head one square to the left, writes a 1, and halts. □

Theorem: BB is not Turing computable.

Proof: For the sake of contradiction, suppose that M is a TM which computes BB. By an earlier theorem, we may assume that M has alphabet $\Sigma = \{\flat, 1\}$. Let M have k states.

> *Claim 1:* $BB(BB(n))$ is computed by a TM, M_2, with $2k$ states.
>
> *Proof*:
>
> 
>
>
> □
>
> *Claim 2:* There is a TM, M_3, with $2k+n$ states and productivity $BB(BB(n))$.

Proof:



□

By claim 2, $BB(2k+n) \geq BB(BB(n))$ for all n. Since BB is increasing, we have $2k+n \geq BB(n)$ for all n. Earlier we showed that $BB(n) \geq 2^{\frac{n}{11}-1}$ for all n. But $2k+n < 2^{\frac{n}{11}-1}$ for large enough n. □

This theorem, combined with Church's thesis, shows that BB is not effectively computable.

The Halting Problem

As an example of a TM that does not halt, consider Fermat's last theorem (FLT): $\forall a, b, c, n > 2,\ a^n + b^n \neq c^n$. Now consider the following effective procedure.

- List out (enumerate) all values $a, b, c, n > 2$.[1]
- For each set of values, check if $a^n + b^n = c^n$.
- If so, then halt; otherwise continue.

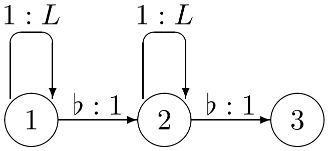
Let M be a TM that does this procedure. (M exists by Church's thesis.) Then M halts on input a blank tape iff FLT is false.

In order to prove the uncomputability of the halting problem, we will represent a TM (with $\Sigma = \{\flat, 1\}$) by a string of symbols written on a tape. We'll use the symbols ♭, 1, L, R, Q, and ',' to "code" a TM.

For example, the TM

[1] To enumerate all 4-tuples $(a, b, c, n > 2)$, use

```
for $i = 8, 9, 10, \ldots$
    list all $(a, b, c, n)$ with $a + b + c + n = i$
```



which computes $n \mapsto n+2$, will be coded by the string

$$\underbrace{q1,1,Lq1}_{f(q_1,1)=(L,q_1)}\ ,\underbrace{q1,\flat,1,q11}_{f(q_1,\flat)=(1,q_2)},\underbrace{q11,1,L,q11}_{f(q_2,1)=(L,q_2)},\underbrace{q11,\flat,1,q111}_{f(q_2,\flat)=(1,q_3)}$$

which gives the values of the transition function. (Note that the string implicitly states that $f(q_3,1) = f(q_3,\flat) =$ undefined.)

Fact: Every TM has such a code.

Fact: There is an effective procedure for determining whether or not a string of symbols codes a TM.

A string in $\{\flat, 1, q, L, R, `,'\}$ can be interpreted as a base 6 integer. Let

$$H(n) = \begin{cases} 2 & \text{if } n \text{ (in base 6) codes a TM} \\ & \text{which halts on input a blank tape} \\ 1 & \text{otherwise} \end{cases}$$

Claim: $H(n)$ is not Turing computable.
I.e., there is no effective procedure which, given a code for a TM, can determine if that TM halts on input a blank tape.

Recursive Functions

Math 260A - Mathematical Logic

October 5, 1988

As we change from Turing computable functions to recursive functions, we'll change our convention regarding the domain of functions. Now we will consider functions over the non-negative integers ($N = \{0, 1, \ldots\}$) instead of over the positive integers. The reason for doing this is historical.

Partial Functions

A k-ary *partial function* f is one whose domain is a subset of N^k and whose co-domain[1] is N; i.e. $f(a_1, \ldots, a_k)$ may be undefined. Notationally, $f(a_1, \ldots, a_k) \uparrow$ means that f is undefined (diverges), and $f(a_1, \ldots, a_k) \downarrow$ means that it is defined (converges).

Every TM, M, computes a partial function, since $f(a_1, \ldots, a_k) \uparrow$ if M, on inputs $a_1 + 1, \ldots, a_k + 1$, either never halts or halts in some non-standard configuration, and $f(a_1, \ldots, a_k) \downarrow$ and equals the output -1 of M on inputs $a_1, \ldots, a_k$ otherwise.

Definition: A partial function is *total* if it converges for all arguments.

Later, we'll show that if f is a k-ary recursive (to be defined) function and $g(a_1, \ldots, a_k) = 1 + f(a_1 - 1, \ldots, a_k - 1)$, then g is Turing computable and conversely.

Composition

Let g be a k-ary partial function and let $h_1, \ldots, h_k$ be k m-ary partial functions. Then f is defined by *composition* from g and $h_1, \ldots, h_k$ if and only if

- f is an m-ary partial function,

[1] If f is a function, then the *image* of f is the set of values of f, and the *co-domain* is the set in which f takes values. For example, consider the function $Z(x) = 0$. Its range is 0, while its co-domain is N.

- $f(a_1, \ldots, a_m) = g(h_1(a_1, \ldots, a_m), \ldots, h_k(a_1, \ldots, a_m))$ if all values are defined,
- $f(a_1, \ldots, a_m) \uparrow$ if $(\exists i \leq k)[h_i(a_1, \ldots, a_m) \uparrow]$, and
- $f(a_1, \ldots, a_m) \uparrow$ if $(\forall i \leq k)[h_i(a_1, \ldots, a_m) \downarrow]$, but $g(h_1(a_1, \ldots, a_k), \ldots, h_k(a_1, \ldots, a_k)) \uparrow$.

This definition is like "call-by-value" because the arguments to g are computed before g is. For example ($k = m = 1$), suppose that $g(a) = 1$ for all a and that $h(a) \uparrow$ for all a. If f is defined by composition from g and h as $f(a) = g(h(a))$, then for all a, $f(a) \uparrow$ (and not $f(a) = 1$).

Minimization

Suppose g is a total $(k+1)$-ary partial function. Then f is defined by minimization from g if and only if

- f is a k-ary partial function, and
- for all $a_1, \ldots, a_k$, $f(a_1, \ldots, a_k) =$ least b such that $g(a_1, \ldots, a_k, b) = 0$ or $f(a_1, \ldots, a_k) \uparrow$ if there is no such b.

Notationally, if f is defined by minimization from g, then $f(a_1, \ldots, a_k) = \mu b[g(a_1, \ldots, a_k, b) = 0]$.

Definition: g is *regular* if for all $a_1, \ldots, a_k$, $\mu b[g(a_1, \ldots, a_k, b) = 0] \downarrow$.

In this case, f is defined by regular minimization from g and will be a total partial function.

Partial Recursive Functions

The *partial recursive functions* (or recursive partial functions) are defined inductively as follows (they are all partial functions):

- The following base functions are all partial recursive:
  - $S(x) = x + 1$
  - $I^m_k(x_1, \ldots, x_m) = x_k$
  - $(x_1, x_2) \mapsto x_1 + x_2$
  - $(x_1, x_2) \mapsto x_1 \cdot x_2$

- $(x_1, x_2) \mapsto x_1 \dot{-} x_2$, where

$$x_1 \dot{-} x_2 = \begin{cases} x - y & \text{if } x > y \\ 0 & \text{if } x \leq y \end{cases}$$

- If $g, h_1, \ldots, h_k$ are partial recursive and f is defined from them by composition, then f is partial recursive.
- If g is partial recursive and f is defined from g by minimization, then f is partial recursive.
- The only functions that are partial recursive are those that are forced to be by the preceding conditions.

Theorem: Let $C_j(a) = j$ be a constant function. Then C_j is partial recursive (and recursive).

Proof: $C_0(a) = a \dot{-} a = I_1^1(a) \dot{-} I_1^1(a)$. (Defined by composition from $\dot{-}$ and I_1^1.)

$$\begin{aligned} C_1(a) &= S(C_0(a)), \\ C_2(a) &= S(C_1(a)), \\ &\vdots \\ C_j(a) &= S(C_{j-1}(a)). \ \square \end{aligned}$$

Definition: The class of *recursive functions* is the smallest class of functions containing $S, I_k^n, +, \cdot,$ and $\dot{-}$, and closed under composition and regular minimization.

Note that every recursive function is a total partial recursive function. Later, we'll show that the total partial recursive functions are the recursive functions.

Examples

The following functions are all recursive:

$$1 \dot{-} x = C_1(x) \dot{-} x$$

$$
\begin{aligned}
\lfloor \sqrt{x} \rfloor &= max(y : y^2 \leq x) \\
&= min(y : (y+1)^2 > x) \\
&= \mu y[(y+1)^2 \dot{-} x > 0] \\
&= \mu y[1 \dot{-} ((y+1)^2 \dot{-} x) = 0]
\end{aligned}
$$

$$
|x - y| = (x \dot{-} y) + (y \dot{-} x)
$$

$$
\begin{aligned}
\lfloor x/y \rfloor &= \begin{cases} 0 & \text{if } y = 0 \\ \lfloor x/y \rfloor & \text{otherwise} \end{cases} \\
&= \mu b[(b+1)y > x \ \vee \ y = 0] \\
&= \mu b[(b+1)y \dot{-} x > 0 \ \vee \ y = 0] \\
&= \mu b[1 \dot{-} ((b+1)y \dot{-} x) = 0 \ \vee \ y = 0] \\
&= \mu b[(1 \dot{-} ((b+1)y \dot{-} x)) \cdot y = 0]
\end{aligned}
$$

$$
x \bmod y = x \dot{-} \lfloor x/y \rfloor
$$

Recursive Functions (cont.)

Math 260A - Mathematical Logic

October 7, 1988

Definition: A k-ary relation (i.e. a subset of N^k) is recursive if and only if its characteristic function is.

Notationally, the characteristic function for relation R is

$$\chi_R(\vec{x}) = \begin{cases} 1 & \text{if } \vec{x} \in R \quad (\text{or } R(\vec{x})) \\ 0 & \text{if } \vec{x} \notin R \quad (\text{or } \neg R(\vec{x})) \end{cases}$$

For example, $x = y$ is a 2-ary relation with characteristic function $\chi_=(x, y) = 1 \dot{-} |x - y|$, so $=$ is recursive.

Theorem: If R and S are k-ary recursive relations, then $R \cup S$, $R \cap S$, and $N^k \setminus R$ are recursive.

Proof:

$$\begin{aligned} \chi_{N^k \setminus R}(\vec{x}) &= 1 \dot{-} \chi_R(\vec{x}), \\ \chi_{R \cap S}(\vec{x}) &= \chi_R(\vec{x}) \cdot \chi_S(\vec{x}), \\ \chi_{R \cup S}(\vec{x}) &= N^k \setminus ((N^k \setminus R) \cap (N^k \setminus S)). \ \Box \end{aligned}$$

Theorem: If S is a k-ary recursive relation and $f_1, \ldots, f_k$ are m-ary recursive functions, then R, defined by $R(\vec{x}) \Leftrightarrow S(f_1(\vec{x}), \ldots, f_k(\vec{x}))$ is a recursive m-ary relation.

Proof: $\chi_R(\vec{x}) = \chi_S(f_1(\vec{x}), \ldots, f_k(\vec{x}))$. $\Box$

Theorem (definition by cases): If $g_1, \ldots, g_k$ are recursive m-ary functions, $R_1, \ldots, R_k$ are recursive m-ary relations, and for all $x_1, \ldots, x_m$ exactly one of $R_i(x_1, \ldots, x_m)$ is true, then the following function is recursive:

$$f(x_1, \ldots, x_m) = \begin{cases} g_1(\vec{x}) & \text{if } R_1(\vec{x}) \\ & \vdots \\ g_k(\vec{x}) & \text{if } R_k(\vec{x}). \end{cases}$$

Proof: $f(x_1, \ldots, x_k) = g_1(\vec{x}) \cdot \chi_{R_1}(\vec{x}) + \ldots + g_k(\vec{x}) \cdot \chi_{R_k}(\vec{x})$. □

Theorem (definition by bounded quantification): Let R be a recursive $k+1$-ary relation, and let g be a recursive k-ary function. Then $S(x_1, \ldots, x_k)$ and $T(x_1, \ldots, x_k)$, as defined next, are both k-ary recursive relations.

$$\begin{aligned} S(x_1, \ldots, x_k) &\Leftrightarrow (\exists x_{k+1} < g(x_1, \ldots, x_k))[R(x_1, \ldots, x_k, x_{k+1})] \\ T(x_1, \ldots, x_k) &\Leftrightarrow (\forall x_{k+1} < g(x_1, \ldots, x_k))[R(x_1, \ldots, x_k, x_{k+1})] \end{aligned}$$

Proof: Consider the following function:

$$f(x_1, \ldots, x_k) = \mu b[R(x_1, \ldots, x_k, b) \vee b = g(x_1, \ldots, x_k)].$$

f is recursive since R, g, and $=$ are recursive and since recursive relations are closed under union.

> *Lemma*: If U is a recursive relation, then $h(x) = \mu b[U(\vec{x}, b)]$ is partial recursive. Furthermore, if for all x there is a b such that $U(\vec{x}, b)$, then h is recursive.
>
> *Proof*: Let $h(\vec{x}) = \mu b[1 \dot{-} \chi_U(\vec{x}, b) = 0]$. This is just the original definition of minimization. □

By the lemma, f is recursive. So,

$$S(x_1, \ldots, x_k) \Leftrightarrow f(x_1, \ldots, x_k) < g(x_1, \ldots, x_k),$$

and

$$\chi_S(x_1, \ldots, x_k) = \begin{cases} 1 & \text{if } f(\vec{x}) < g(\vec{x}) \\ 0 & \text{if } f(\vec{x}) \geq g(\vec{x}) \text{ (in this case, } f(\vec{x}) = g(\vec{x})). \end{cases}$$

So χ_S and S are recursive. We can proceed the same way for T or we can observe that

$$T(x_1, \ldots, x_k) \Leftrightarrow \neg(\exists x_{k+1} < g(x_1, \ldots, x_k))[\neg R(x_1, \ldots, x_k), x_{k+1})]. \ \Box$$

Applications

Theorem: The following sets are recursive:

1. $\{(x, y) : x|y\}$

   Proof: $x|y \Leftrightarrow (\exists z < y + 1)[xz = y]$. $\Box$

2. $\Pr x = \{x : x \text{ is prime}\}$

   Proof: $\Pr x \Leftrightarrow (\forall z < x)[(z = 1 \vee z \not| x) \wedge x \neq 0]$. $\Box$

   (Note that $(\forall z < 0)[R(z)]$ is trivially true no matter what R is.)

3. $PP(x, y) = \{(x, y) : \Pr x \wedge y \text{ is a power of } x\}$

   Proof:

   $$\begin{aligned} PP(x,y) &\Leftrightarrow \Pr x \wedge (\forall z < y + 1)[(z|y \wedge \Pr z) \to z = x] \\ &\Leftrightarrow \Pr x \wedge (\forall z < y + 1)[z \not| y \vee \neg \Pr z \vee z = x] \end{aligned}$$

   or

   $$\begin{aligned} PP(x,y) &\Leftrightarrow \Pr x \wedge (\forall z < y + 1)[z|y \to x|z \vee z = 1] \\ &\Leftrightarrow \Pr x \wedge (\forall z < y + 1)[z \not| y \vee x|z \vee z = 1] \end{aligned}$$

4. $\{(x, y) : y \text{ is a power of } x\}$

   Proof: HARD! We'll do it later when we have more tools.

Primitive Recursion

The exponentiation relation above is hard to define in terms of the methods we currently have for generating recursive functions. But x^y has the following simple "recursive" definition:

$$\begin{aligned} x^0 &= 1, \\ x^{y+1} &= x \cdot x^y. \end{aligned}$$

This form of definition is called definition by *primitive recursion.*

Definition: If g is a k-ary function and h is a $k+2$-ary function, then f is defined by primitive recursion from g and h if f is a $k+1$-ary function and

$$\begin{aligned} f(x_1, \ldots, x_k, 0) &= g(x_1, \ldots, x_k) \\ f(x_1, \ldots, x_k, m+1) &= h(x_1, \ldots, x_k, m, f(x_1, \ldots, x_k, m)) \end{aligned}$$

In this definition, we allow for the possibility of $k = 0$ so that g can be a constant. For example, $x!$ is defined by setting g to the constant function 1 and h to the product function, since $0! = 1$ and $(x+1)! = (x+1) \cdot x!$. Later, we'll see that functions defined by primitive recursion are recursive, so that exponentiation is recursive.

Primitive Recursive Functions

Math 260A - Mathematical Logic

October 10, 1988

Review of Quantification

$$\begin{array}{rcl}
\forall x & \text{means} & \text{“for all” } x \\
\exists x & \text{means} & \text{“there exists” } x
\end{array}$$

$$\begin{array}{rcl}
(\forall x \leq y)[\ldots] & \text{means} & (\forall x)[\text{if } x \leq y \text{ then } \ldots] \\
 & \text{or} & (\forall x)[x \leq y \to \ldots] \\
 & \text{or} & (\forall x \in N)[x \leq y \to \ldots] \\
 & \text{or} & (\forall x)[x \in N \wedge x \leq y \to \ldots] \\
 & & \text{(note also that “if } x \leq y \text{ then } \ldots\text{”} \\
 & & \text{is equivalent to “}\neg(x \leq y) \vee \ldots\text{”)}
\end{array}$$

$$\begin{array}{rcl}
(\exists x \leq y)[\ldots] & \text{means} & (\exists x)[x \in N \wedge x \leq y \wedge \ldots]
\end{array}$$

$$\begin{array}{rcl}
\neg(\exists x \leq y)[x = 1] & \equiv & (\forall x \leq y)[x \neq 1] \\
 & \equiv & \neg(\exists x)[x \leq y \wedge x = 1] \\
 & \equiv & (\forall x)[\neg(x \leq y \wedge x = 1)] \\
 & \equiv & (\forall x)[\neg(x \leq y) \vee x \neq 1)] \\
 & \equiv & (\forall x)[(x \leq y) \to x \neq 1)] \\
 & \equiv & (\forall x \leq y)[x \neq 1)]
\end{array}$$

Primitive Recursive Functions

Definition: The primitive recursive functions are the smallest class of functions on N which contain

$$\begin{aligned} Z(x) &= 0 \\ S(x) &= x+1 \\ id_k^m(x_1,\ldots,x_m) &= x_k,\ \forall m \geq k \geq 0 \end{aligned}$$

and are closed under primitive recursion and composition.

Primitive recursion seems more natural than minimization, but it turns out that primitive recursion is much less powerful.

Theorem: If g and h are k- and $k+2$-ary recursive functions (not primitive recursive) and if f is defined from g and h by primitive recursion, then f is also recursive.

The proof of this is more longwinded than any we've done so far and will require some preliminary theorems.

Corollary: Every primitive recursive function is recursive. (Since the base functions are recursive.)

So, $x!$, x^y, etc. are recursive.

Definition: The BJ-recursive functions (Boolos & Jeffrey) are the smallest class of functions containing Z, S, and id_k^m and closed under composition, primitive recursion, and minimization.

Corollary: recursive = BJ-recursive.

Definition: A relation (a subset of N^k) is primitive recursive if and only if its characteristic function is.

Theorem: If g is a k-ary primitive recursive function and R is a $k+1$-ary primitive recursive relation, then $(\exists a_{k+1} \leq g(a_1,\ldots,a_k))[R(a_1,\ldots,a_{k+1})]$ is a primitive recursive relation. Similarly for $\forall$.

Examples

Addition is primitive recursive since

$$\begin{aligned} x+0 &= x, \\ x+(y+1) &= S(x+y). \end{aligned}$$

Multiplication is primitive recursive since

$$\begin{aligned} x \cdot 0 &= 0, \\ x \cdot (y+1) &= x \cdot y + x. \end{aligned}$$

Exponentiation is primitive recursive since

$$\begin{aligned} x^0 &= 1 \\ x^{y+1} &= x \cdot x^y. \end{aligned}$$

Super-exponentiation is primitive recursive since

$$\begin{aligned} x \Uparrow 0 &= 1 \\ x \Uparrow (y+1) &= x^{x \Uparrow y}. \end{aligned}$$

Last time, we saw that it was hard to show that exponentiation is recursive. So let's start with a simpler version.

Definition:

$$p \uparrow y = \begin{cases} p^y & \text{if } \mathrm{Pr}\, p \wedge y \le p \\ 0 & \text{otherwise} \end{cases}$$

Theorem: $p \uparrow y$ is recursive.

Lemma: $\frac{p^n-1}{p-1} \equiv n \pmod{p-1}$.

Proof:

$$\begin{aligned} \frac{p^n-1}{p-1} &= p^{n-1} + p^{n-2} + \cdots + p^2 + p + 1 \\ p &\equiv 1 \pmod{p-1} \\ p^j &\equiv 1 \pmod{p-1} \\ \frac{p^n-1}{p-1} &\equiv n \pmod{p-1} \; \square \end{aligned}$$

Proof: Let

$$f(x,y) = \begin{cases} x^y & \text{if } \mathrm{Pr}\, x \wedge y < x-1 \\ 0 & \text{otherwise} \end{cases}$$

Then

$$\begin{aligned} f(x,y) &= \mu b[(\frac{b-1}{x-1} \equiv y \pmod{x-1} \wedge b \text{ is a power of } x) \vee \\ &\quad (\neg \operatorname{Pr} x \vee y \geq x-1)] \end{aligned}$$

which is recursive. (Note that $rem(\frac{\lfloor \frac{b-1}{x-1} \rfloor}{x-1}) = y$.) So

$$x \uparrow y = \begin{cases} f(x,y) & \text{if } y < x-1 \\ x \cdot f(x, x-2) & \text{if } y = x-1 \\ x^2 \cdot f(x, x-2) & \text{if } y = x \\ 0 & \text{if } y > x. \end{cases} \qquad \square$$

Let p_a be the $(a+1)^{st}$ prime; e.g.

$$\begin{aligned} p_0 &= 2 \\ p_1 &= 3 \\ p_2 &= 5 \\ p_3 &= 7 \\ &\vdots \end{aligned}$$

Theorem: $f(x) = p_x$ is recursive.
Proof: Let $NP(x) =$ the "next prime" after x; i.e.

$$NP(x) = \mu p[\operatorname{Pr} p \ \wedge \ p > x].$$

Define g as follows:

$$\begin{aligned} g(0) &= 2^0 = 1 \\ g(1) &= 2^0 \cdot 3^1 = 3 \\ g(2) &= 2^0 \cdot 3^1 \cdot 5^2 = 75 \\ g(3) &= 2^0 \cdot 3^1 \cdot 5^2 \cdot 7^3 = 25725 \\ &\vdots \\ g(a+1) &= g(a) \cdot (p_{a+1})^{a+1} \end{aligned}$$

g is recursive since

$$\begin{aligned}
g(x) \quad = \quad & \mu b[2 \nmid b \wedge (\forall \text{primes } p < b+2)[\\
& (p|b \vee p = 2) \rightarrow (\exists i < p)[\\
& (p \uparrow i)|b \wedge p \uparrow (i+1) \nmid b\\
& \wedge(i < x \rightarrow ((NP(p) \uparrow (i+1))|b \wedge (NP(p) \uparrow (i+2)) \nmid b))]]].
\end{aligned}$$

So,

$$\begin{aligned}
f(x) = p_x \quad = \quad & \mu p[(\Pr p \wedge p|g(x) \wedge NP(p) \nmid g(x)) \vee\\
& (x = 0 \wedge p = 2)]. \ \square
\end{aligned}$$

Note that in the definition of g, “$(p \uparrow i)|b \wedge p \uparrow (i+1) \nmid b$” means that p is in the prime factorization of b and that “$NP(p) \uparrow (i+2) \nmid$” means that $NP(p) \uparrow (i+1)$ is in the prime factorization of b.

Primitive Recursive → Recursive Functions

Math 260A - Mathematical Logic

October 12, 1988

We want to see how primitive recursion can be used to define recursive functions. To do this, we have to introduce a new topic.

Sequence Coding

Sequence coding uses a single integer to code a finite sequence of integers. An integer of the following form

$$w = (p_0)^{a_0+1} \cdot (p_1)^{a_1+1} \cdot \ldots \cdot (p_n)^{a_n+1}$$

will code a sequence $a_0, a_1, \ldots, a_n$. Note that every sequence has a unique such code. But, not every integer codes a sequence; e.g. the odd integers. The empty sequence is coded by 1. The '+1's are included in the exponents of the coding so that we can distinguish between sequences with a different number of zeroes on the right and otherwise the same.

The relation $Seq(x) \Leftrightarrow$ "x codes a sequence" is recursive since

$$Seq(x) \Leftrightarrow (\forall \text{primes } p, q \leq x)[(q < p \wedge p|x) \rightarrow q|x].$$

The function $lh(x) =$ the number of elements in the sequence coded by x if x codes a sequence or 0 if x doesn't code a sequence is recursive since

$$lh(x) = \mu b[p_b \not| x \vee \neg Seq(x)].$$

Now we want to define $\beta(n,x) = n^{th}$ element in the sequence coded by x, where $n = 0$ represents the first element; i.e. $\beta(n, (p_0)^{a_0+1} \cdot \ldots \cdot (p_n)^{a_n+1}) = a_n$. Because this is a little hard to do right now, we'll instead use a restricted function $\beta^-(n,x) = min(\beta(n,x), p_{n-1})$. β^- is recursive since

$$\beta^-(n,x) = \mu i[i = p_{n-1} \vee (p_n \uparrow (i-2)) \not| x].$$

This definition of β^{-} allows us to handle sequences $a_0 \ldots a_n$ as long as $a_i < p_i$ for all i.

Notation: $\langle a_0 \ldots a_n \rangle$ will be defined to be the integer coding the sequence $a_0 \ldots a_n$; i.e.

$$\langle a_0 \ldots a_n \rangle = 2^{a_0+1} \cdot 3^{a_1+1} \cdot \ldots \cdot (p_n)^{a_n+1}.$$

This is the Gödel number of the sequence.

Now we are ready to prove the unproven theorem from last lecture.

Theorem (restated from last lecture): If g and h are recursive and f is defined from them by primitive recursion, then f is recursive.

Proof: The idea is to define the sequence

$$\bar{f}(\vec{x}, m) = \langle f(\vec{x}, 0), f(\vec{x}, 1), \ldots, f(\vec{x}, m) \rangle,$$

and ensure that

$$\begin{aligned} f(\vec{x}, 0) &= g(\vec{x}), \text{ and} \\ f(\vec{x}, i+1) &= h(\vec{x}, i, f(\vec{x}, i)). \end{aligned}$$

First, we'll find the Gödel number of a sequence of the form

$$\langle 0, 0, \ldots, 0, f(\vec{x}, 0), \ldots, f(\vec{x}, m) \rangle$$

(which $= \beta(i, x)$); i.e.

$$\begin{aligned} F(\vec{x}, m) &= \mu b[Seq(b) \wedge (\exists i < b)[\beta^{-}(i, b) = g(\vec{x}) \wedge lh(b) = i + m + 1 \wedge \\ &\quad (\forall j < m)[\beta^{-}(i + j + 1, b) = h(\vec{x}, j, \beta^{-}(i + j, b)]]]. \end{aligned}$$

F is recursive since it is defined by regular minimization. So

$$f(\vec{x}, m) = \beta^{-}(lh(F(\vec{x}, m)) \dot{-} 1, F(\vec{x}, m)). \; \Box$$

Corollaries:

- Exponentiation is recursive.

- By using p^n instead of $p \uparrow n$, we can define the unrestricted $\beta(n, x)$ recursively.

- $\langle a_0, \ldots, a_n \rangle * \langle b_0, \ldots, b_m \rangle = \langle a_0, \ldots, a_n, b_0, \ldots, b_m \rangle$ is recursive. ($x * y = 0$ if either x or y does not code a sequence.

Fact: Every function and predicate that we have proved to be recursive is actually primitive recursive.

Partial Recursive Functions are Turing Computable

Math 260A - Mathematical Logic

October 14, 1988

Theorem: If f is a partial recursive function, then there is a Turing machine, M, such that for all $n \geq 0$,

1. if $f(n)\downarrow$, then M on input $n+1$ in standard notation halts and outputs $f(n)+1$ in standard notation, and

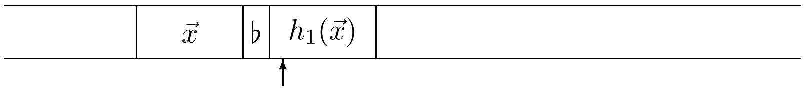
2. if $f(n)\uparrow$, then M on input $n+1$ in standard notation never halts.

Proof (by induction on the complexity of the partial recursive definition of f): We will actually prove a stronger result; i.e. that there is such an M which never moves to the left of the starting position, and that there is such an M for k-ary functions, $k \geq 1$.

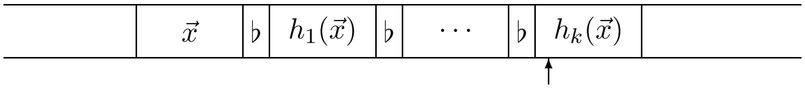
Basis: Easy but somewhat tedious for $S, +, \cdot, \dot{-}, id_k^m$.

Induction:

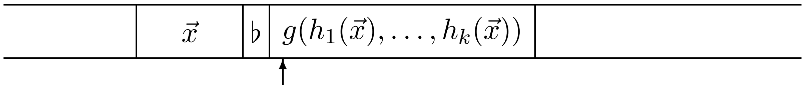
1. If it holds for $g, h_1, \ldots, h_k$, then it holds for $f(\vec{x}) = g(h_1(\vec{x}), \ldots, h_k(\vec{x}))$. Assume that the machines for $g, h_1, \ldots, h_k$ are $M_g, M_{h_1}, \ldots, M_{h_k}$. M will incorporate all of these internally. First, M copies $\vec{x}$ to the right of its input and runs M_{h_1} on the copy. This leaves the tape in the following configuration:

   Now M copies $\vec{x}$ to the right of $h_1(\vec{x})$ and runs M_{h_2} on that copy. M continues in this way until $h_k(\vec{x})$ has been computed, and the tape looks like:

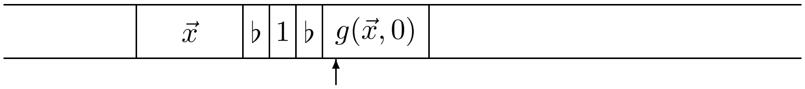


At this point, M moves the tape head back to leftmost square of $h_1(\vec{x})$ and runs M_g to yield the following configuration:

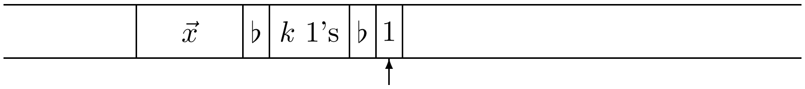


Finally, M shifts $g(h_1(\vec{x}), \ldots, h_k(\vec{x}))$ back to the starting square, erases the rest of the tape, and halts with the head at the leftmost square of the output. The preceding construction works fine if all the intermediate functions converge. If any of them diverge, then $f \uparrow$, and M doesn't halt.

2. If it holds for g, then it holds for $f(\vec{x}) = \mu y[g(\vec{x}, y) = 0]$. M first writes a 1 to the right of its input. This will be a counter which represents the current value of y that we are using for computing g. M now copies $\vec{x}$ and the counter to the right and runs M_g. (Note that by the definition of minimization, g is total.) This leaves the output tape in the following configuration:

   Suppose that $g(\vec{x}, 0) > 0$. Then M erases $g(\vec{x}, 0)$, goes back to the counter, increments it to 2 (remember, a string of i 1's represents the number $i - 1$), and repeats the above step; i.e. it copies $\vec{x}$ and the counter to the right and runs M_g. Now, suppose that after k iterations of this process, M finds that $g(\vec{x}, k - 1) = 0$. The tape will then look like:

   Now, M shifts the counter back to the starting position, erases the rest of the tape, and returns the head to the starting position. If there is no y such that $g(\vec{x}, y) = 0$, then the above process will iterate forever; which is what we want, since in that case, $f \uparrow$. □

Note that $\mu y[g(\vec{x}, y) = 0]$ is a valid definition of a partial recursive function only if g is total. But there is no general way of knowing whether or not g is total, so we don't have an effective way of building partial recursive functions. We could have defined minimization as

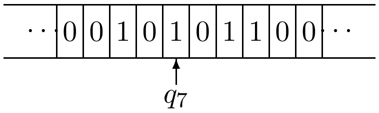
$$f(\vec{x}) = \mu y[g(\vec{x}, y) = 0 \wedge (\forall z < y)[g(\vec{x}, z) \downarrow]].$$

Then, even if g is not total, f is partial recursive.

Turing computable functions are Partial Recursive

In order to show this, we need a Gödel numbering of Turing machine computations. In this development, we'll let 0 take the place of ♭ so that we can think of tape squares as binary numerals. We'll also restrict the Turing machines to the alphabet $\{0, 1\}$.

Given a configuration of a Turing Machine, e.g.



we will code it as the triple of integers $\langle i, \ell, r\rangle$ where i is the state number (e.g. 7), ℓ is the tape contents to the left of the tape head interpreted as a binary integer (e.g. 2), and r is the tape contents under and to the right of the tape head interpreted as a binary number in reverse (e.g. 13). The code for the above configuration is

$$\langle 7, 2, 13\rangle = 2^8 \cdot 3^3 \cdot 5^{14}.$$

The code tells us everything about the current configuration, but says nothing about how we got there. A computation, or series of configurations, is coded by the Gödel number of a sequence:

$$\langle\langle i_0, \ell_0, r_0\rangle,\ \langle i_1, \ell_0, r_0\rangle, \ldots, \langle i_n, \ell_n, r_n\rangle\rangle.$$

This codes a computation of n steps.

Another function that we'll need is $NEXT_M$ which, given a tape configuration, finds the next configuration that M would go to; i.e.

$$NEXT_M(\langle i,\ell,r\rangle) = \begin{cases} \langle i^{'},\ell^{'},r^{'}\rangle & \text{(the configuration of } M \\ & \text{one step after } \langle i,\ell,r\rangle \\ 0 & \text{if } M \text{ halts at } \langle i,\ell,r\rangle \end{cases}$$

$NEXT_M$ is defined in terms of the functions $NEXTQ_M$ which gives the next state, $NEXTL_M$ which gives the next left half of the tape, and $NEXTR_M$ which gives the next right half of the tape. These functions are defined as follows:

$$\begin{aligned} NEXTQ_M(i, r \bmod 2) &= \text{definition by cases according to } M \\ NEXTL_M(i, r \bmod 2, \ell) &= \begin{cases} \ell & \text{if tape head doesn't move} \\ \ell/2 & \text{if tape head moves left} \\ 2\ell + (r \bmod 2) & \text{if tape head moves right} \end{cases} \\ NEXTR_M(i, r \bmod 2, \ell \bmod 2, r) &= \begin{cases} r & \text{if tape head doesn't move} \\ 2r + (\ell \bmod 2) & \text{if tape head moves left} \\ r/2 & \text{if tape head moves right} \\ 2\lfloor r/2 \rfloor & \text{if tape head writes a 0} \\ 2\lfloor r/2 \rfloor + 1 & \text{if tape head writes a 1} \end{cases} \end{aligned}$$

Note that $r \bmod 2$ is the symbol under the tape head, and $\ell \bmod 2$ is the symbol to the left of the tape head.

Turing Computable Functions are Partial Recursive

Math 260A - Mathematical Logic

October 17, 1988

Last time, we defined the function

$$NEXT_M(\langle i,\ell,r\rangle) = \begin{cases} \langle i^{'},\ell^{'},r^{'}\rangle & \text{(the configuration of } M \\ & \text{one step after } \langle i,\ell,r\rangle\text{)} \\ 0 & \text{if } M \text{ halts at } \langle i,\ell,r\rangle \end{cases}$$

which determined the next configuration of a Turing machine from the current configuration. Now we want to describe

$$\langle\langle i_0,\ell_0,r_0\rangle \ldots \langle i_n,\ell_n,r_n\rangle\rangle$$

which codes a computation of M. Let $NEXT_e(\langle i,\ell,r\rangle) = NEXT_M(\langle i,\ell,r\rangle)$ where $e \in N$ is the e^{th} Turing machine (in a sequence without gaps; i.e. every e corresponds to some Turing machine). Then we want relation $T_e(x,w)$ to represent the fact that w codes a halting computation of the e^{th} Turing machine when started on input $x+1$. (Note that w is a sequence of tuples; the first is the starting configuration, and the last is the halting configuration.)

$$\begin{aligned} T_e(x,w) \quad \Leftrightarrow \quad & Seq(w) \wedge \\ & \beta(0,w) = \langle 1,0,2^{x+2}-1\rangle \wedge \\ & (\forall i < lh(w)\dot{-}1)[\beta(i+1,w) = NEXT_e(\beta(i,w))] \wedge \\ & \beta(lh(w)\dot{-}1,w) \neq 0 \wedge \\ & NEXT_e(\beta(lh(w)\dot{-}1,w)) = 0. \end{aligned}$$

$\beta(0,w) = \langle 1,0,2^{x+2}-1\rangle$ represents the fact that the starting state is 1, the tape to the left of the head is blank, and there are $x+1$ 1's as input. $\beta(lh(w)\dot{-}1,w) \neq 0$ eliminates the cases where w is padded on the right with

0's; this doesn't matter since we'll use a minimization operator to define the Turing machine code for a partial recursive function, but it doesn't hurt to specify it exactly what we mean.

Fact:

$$T(e,x,w) \Leftrightarrow T_e(x,w)$$

is also primitive recursive, but is a lot harder to define. $T(e,x,w)$ is known as the Kleene T predicate.

Now, in addition to the code for a computation, we might also want to get the answer of a Turing machine computation:

$$\mathit{Result}(w) = \log_2[\beta(2, \beta(lh(w) \dot{-} 1, w)) + 1] - 1.$$

The inside β is the last tuple in w. The outside β is the right-hand side of the output tape. For example, suppose a Turing machine writes out five 1's for outputting 4. Then

$$\begin{aligned} r_n &= 11111_2 \\ r_n + 1 &= 100000_2 = 2^5 \\ \log_2(r_n + 1) &= 5. \end{aligned}$$

(Note that it is not hard to show that $\log_2$ is primitive recursive.)

Theorem: Suppose that f is partial recursive. Then there is an integer e (for the e^{th} Turing machine) such that for all x,

$$f(x) = \mathit{Result}(\mu w[T_e(x,w)]).$$

Proof: We have already shown that there is a Turing machine, M, such that for all x, M on input $x+1$ 1's either outputs $f(x)+1$ 1's in standard notation or never halts if $f(x)\uparrow$. Let M be the e^{th} Turing machine. □

Kleene Normal Form

Theorem: For every partial recursive function f, there are primitive recursive functions g and h such that

$$f(\vec{x}) = g(\mu y[h(\vec{x}, y) = 0]).$$

Proof: Let $h = 1 \dot{-} \chi_{T_e}$. □

This theorem is important because it says that every partial recursive function can be expressed in terms of a single use of minimization. (Note that you can convert any partial recursive to its normal form by first building a Turing machine to compute it.)

Theorem: Every total partial recursive function is recursive.

Proof: Let f be a total partial recursive function, and let g and h be as above. g and h are total since they are primitive recursive. For all $\vec{x}$, there is a y such that $h(\vec{x}, y) = 0$; otherwise, $f(\vec{x}) \uparrow$. So $\mu y[h(\vec{x}, y) = 0]$ is regular minimization, and so f is recursive. □

Theorem (definition by cases): If g and h are partial recursive functions and R is a recursive relation, then

$$f(x) = \begin{cases} g(x) & \text{if } R(x) \\ h(x) & \text{otherwise} \end{cases}$$

is partial recursive.

Proof: Homework. (Hint: use Kleene normal form. You can also use induction on the complexity of partial recursive functions, but it's not as clear. Note that our earlier proof for the recursive counterpart of this theorem doesn't work here.)

Primitive Recursive $\neq$ Recursive

We have already shown that primitive recursive functions are recursive. (We simulated primitive recursion using Gödel numbers of sequences.) Now we want to show that not every recursive function is primitive recursive. Just as we found a function (the busy beaver function) which grew too fast to be recursive, we'll show that there is a function (Ackermann's function) which is recursive, yet grows too fast to be primitive recursive.

Definition:

$$f^n(x) = \underbrace{f(f(\cdots f}_{n\ f\text{' s}}(x)\cdots)).$$

For example,

$$\begin{aligned} f^0(x) &= x \\ f^1(x) &= f(x) \end{aligned}$$

and in general,

$$f^{i+1}(x) = f(f^i(x)).$$

Now we'll define a family of fast growing functions.

Definition:

$$\begin{array}{rcl} F_0(x) & = & x+1 \\ F_{i+1}(x) & = & F_i^{x+1}(x) \end{array}$$

Proposition:

$$\begin{array}{rclcl} F_0(x) & = & x+1, & & \\ F_1(x) & = & F_0^{x+1}(x) & = & 2x+1, \\ F_2(x) & = & F_1^{x+1}(x) & = & 2^{x+1}x + 2^x + 2^{x-1} + \ldots + 1 \\ & & & = & 2^{x+1}x + 2^{x+1} - 1 > 2^x, \\ F_3(x) & = & F_2^{x+1}(x) & > & 2 \Uparrow x. \end{array}$$

$F_4(x)$ grows faster, $F_5(x)$ grows even faster, etc.

Proposition: Each F_n is primitive recursive.

Proof: By induction on n.

Definition: A function g eventually dominates h if $\exists n[(\forall m > n)[g(m) > h(m)]]$.

We will construct a function $A(n)$ (Ackermann's function) which eventually dominates every primitive recursive function. $A(n)$ will be built by diagonalization over the F's, and will be recursive.

Primitive Recursive $\subset$ Recursive

Math 260A - Mathematical Logic

October 19, 1988

Last time, we defined a family of functions,

$$\begin{aligned} F_0(x) &= x+1 \\ F_{i+1}(x) &= F_i^{x+1}(x) \end{aligned}$$

We want to use these functions to define the function $A(n)$ which is recursive, but not primitive recursive. To do this, we first have to prove a few lemmas.

Lemma:

1. $\forall i, \forall x,\ F_i(x) > x$.
2. $\forall i > 1, \forall x > 0,\ F_i(x) > F_{i-1}(x)$.[1]

Proof: We'll prove (1) by induction on i and (2) will just happen along the way.

Basis: $i = 0$. (1) is obvious because $F_0(x) = x + 1 > x$.

Induction: Suppose (1) is known for values smaller than i. Then

$$\begin{array}{rcll} F_i(x) & = & F_{i-1}^{x+1}(x) & \text{by definition} \\ & > & F_{i-1}^{x}(x) & \text{by induction} \\ & > & F_{i-1}^{x-1}(x) & \text{by induction} \\ & \vdots & & \vdots \\ & > & F_{i-1}(x) & \text{by induction} \\ & > & x & \text{by induction } x+1 \text{ times } \square \end{array}$$

[1] Note that we have $\forall i > 1$. For $i = 1$ and $x = 0$, we have $F_1(0) = F_0(0) = 1$.

Question: What is $F_i(0)$?

Answer: $F_i(0) = F_{i-1}(0) = \ldots = F_0(0) = 1$.

Lemma: F_i is strictly increasing; i.e. for all $i, x, y \geq 0$, if $x > y$ then $F_i(x) > F_i(y)$.

Proof: by induction on i.

Basis: $i = 0$. Obvious since if $x > y$ then $x + 1 > y + 1$.

Induction: Suppose it is known for smaller values of i. Then

$$\begin{array}{rcll} F_i(x) & = & F_{i-1}^{x+1}(x) & \text{by definition} \\ & > & F_{i-1}^{x}(x) & \text{by (1) above} \\ & > & F_{i-1}^{x-1}(x) & \text{by (1) above} \\ \vdots & & & \vdots \\ & > & F_{i-1}^{y+1}(x) & \text{by (1) above.} \end{array}$$

Now, $F_i(y) = F_{i-1}^{y+1}(y)$ by definition, so

$$\begin{array}{rcll} F_{i-1}(x) & > & F_{i-1}(y) & \text{by induction} \\ F_{i-1}^{2}(x) & > & F_{i-1}^{2}(y) & \text{by induction} \\ & \vdots & & \vdots \\ F_{i-1}^{y+1}(x) & > & F_{i-1}^{y+1}(y) & \text{by induction } \Box \end{array}$$

Definition: Let g be a unary function, and let h be a $k-$ary function. g *eventually dominates* h if and only if there exists an $n \geq 0$ such that $g(max\{x_1, \ldots, x_k\}) > h(x_1, \ldots, x_k)$ whenever $max\{x_1, \ldots, x_k\} > n$. Equivalently, g eventually dominates h if and only if $g(max\{x_1, \ldots, x_k\}) > h(x_1, \ldots, x_k)$ for all but finitely many $x_1, \ldots, x_k$.

Lemma: If g is increasing, then g eventually dominates h if and only if there exists an $n \geq 0$ such that $g(max\{x_1, \ldots, x_k, n\}) > h(x_1, \ldots, x_k)$ for all $x_1, \ldots, x_k$.

Theorem: Every primitive recursive function is eventually dominated by some F_i. (Recall that the F_i's are all primitive recursive.)

Proof: by induction on the definition of primitive recursive functions.

Basis: $S(x), id_k^n$, and $Z(x)$ are all dominated by F_1.

Induction: We need to consider two cases:

Composition Let $g, h_1, \ldots, h_k$ satisfy the theorem, and let f be defined by composition from them. We need to show that f satisfies the theorem. Since $g, h_1, \ldots, h_k$ satisfy the theorem, suppose that F_i eventually dominates $g, h_1, \ldots, h_k$; i.e.

$$\begin{array}{lcll} F_i(max\{\vec{x}, n\}) & > & g(\vec{x}) & \text{for all } \vec{x} \\ F_i(max\{\vec{y}, n\}) & > & h_j(\vec{y}) & \text{for all } \vec{y}, j \end{array}$$

Then,

$$\begin{array}{rcll} f(\vec{x}) & = & g(h_1(\vec{y}), \ldots, h_k(\vec{y})) & \text{by defn of composition} \\ & < & F_i(max\{h_1(\vec{y}), \ldots, h_k(\vec{y}), n\}) & \text{by induction} \\ & \leq & F_i(max\{F_i(max\{\vec{y}, n\}), n\}) & \text{by induction} \\ & \leq & F_i(F_i(max\{\vec{y}, n\})) & \\ & = & F_i^2(max\{\vec{y}, n\}) & \\ & < & F_{i+1}(max\{\vec{y}\}) & \text{when } max\{\vec{y}\} \text{ is} \\ & & & \text{large enough } (\geq 2) \end{array}$$

So f is eventually dominated by F_{i+1}.

(Intuitively, since g and the h_j's are eventually dominated by F_i, $g(h_1(-), \ldots, h_k(-))$ is eventually dominated by F_i^2 which is eventually dominated by F_{i+1}.)

Primitive Recursion Let g and h satisfy the theorem, and suppose that F_i eventually dominates them; i.e.

$$\begin{array}{lcll} F_i(max\{\vec{x}, n\}) & > & g(\vec{x}) & \text{for all } \vec{x} \\ F_i(max\{\vec{x}, m, y, n\}) & > & h(\vec{x}, m, y) & \text{for all } \vec{x}, m, y \end{array}$$

We need to show that if f is defined by primitive recursion from g and h, then

$$f(m, \vec{x}) \leq F_i^{m+1}(max\{m, \vec{x}, n\}) < F_{i+1}(max\{\vec{x}, n\}).$$

(Note that we could have used $max\{\vec{x}, n\}$ instead of $max\{m, \vec{x}, n\}$; the m isn't necessary since we're applying F_i $m+1$ times and are guaranteed to be greater than m.) We will prove the above inequality by induction on n.

$$\begin{array}{rcl} f(0,\vec{x}) & = & g(\vec{x}) \\ & \leq & F_i(max\{0,\vec{x},n\}) \end{array}$$

$$\begin{array}{rcll} f(m+1,\vec{x}) & = & h(\vec{x},m,f(m,\vec{x})) & \\ & < & F_i(max\{\vec{x},m,f(m,\vec{x})\}) & \text{by ind.} \\ & \leq & F_i(max\{\vec{x},m,F_i^{m+1}(max\{m,\vec{x},m,n\}),n\}) & \text{by ind.} \\ & = & F_i^{m+2}(max\{m,\vec{x},n\}) & \\ & \leq & F_i^{m+2}(max\{m+1,\vec{x},n\}) & \end{array}$$

So f is eventually dominated by F_{i+1}. $\Box$

Definition: $A(n) = F_n(n)$. This is similar to Ackermann's original function.

Theorem: A(n) eventually dominates F_i for all i.
Proof: For $n > i$, $A(n) > F_i(n)$. $\Box$

Theorem: $A(n)$ is not primitive recursive.
Proof: $A(n)$ is not dominated by any F_i. $\Box$

Theorem: $A(n)$ is recursive.
Proof: via Church's Thesis. Let $g(x,y,z) = F_x^y(z)$. So g is effectively computable. So $A(n) = g(n,1,n)$ is also effectively computable and hence is recursive. $\Box$

Note that g above is not primitive recursive; it's "doubly recursive".[2] Homework: prove that g is recursive without using Church's Thesis.

[2] *Double recursion* means that we do recursion on (x,y) ordered lexicographically; i.e., $(x_1,y_1) < (x_2,y_2)$ iff $x_1 < x_2$, or $x_1 = x_2$ and $y_1 < y_2$. Using this, a possible definition for g is

$$g(x,y,z) = \begin{cases} z & \text{if } y = 0 \\ z+1 & \text{if } y = 1 \text{ and } x = 0 \\ g(x,1,g(x,y-1,z)) & \text{if } y > 1 \\ g(x-1,z+1,z) & \text{if } y = 1 \text{ and } x > 0 \end{cases}$$

Theorem: There is a $0/1$ valued function which is recursive but not primitive recursive.

Proof: Homework. Hint: look at techniques in Boolos & Jeffrey. Suggestion: Let $B(n, i) = i^{th}$ bit in the binary representation of $A(n)$; this may be primitive recursive.

> *Question*: What about $C(n) = B(n, 0) = A(n) \bmod 2$?
>
> *Answer*: There might be some easy way determining whether or not $A(n)$ is even or odd, and similarly for other bits. In general,
>
> $$R(n, x) \Leftrightarrow x = A(n)$$
>
> is primitive recursive. To see this, note that the computation of R is bounded by x, so you stop computing $A(n)$ when you get to a number bigger than x.

Theorem: A function is primitive recursive if there exists a Turing machine that computes it whose run-time is bounded by one of the F_i's.

Proof: next time.

Computation Time

Math 260A - Mathematical Logic

October 21, 1988

Definition: Let M be a Turing machine whose alphabet, Σ, includes $\flat$. Let x be a word over Σ without $\flat$'s; i.e. $x \in (\Sigma \setminus \{\flat\})^*$. The run-time of M on input x is n if and only if M, when started on x in standard notation, halts after exactly n steps.

Theorem: The following are equivalent:

1. f is primitive recursive.
2. There is a Turing machine, M, and an $i \geq 0$, such that for all n, M, on input $n+1$ 1's, has run-time $\leq F_i(n+1)$ and outputs $f(n)+1$ 1's. (Note: if i is large enough, then $F_i(1)$ is arbitrarily large and can handle cases with finitely many exceptions.)

Proof:

1 ⇒ 2 (Easy direction) The proof is by induction on the complexity of the definition of primitive recursive functions. (A proof similar to last lecture's proof can actually prove a more general statement for $k-$ary functions.)

2 ⇒ 1 Let f be computed by M, the e^{th} Turing machine, with run-time $\leq F_i(n+1)$; i.e. for all x,

$$f(x) = \mathit{Result}(\mu w[T_e(x,w)]).$$

To make f primitive recursive, we have to bound w. How big *is* w? Well,

$$\begin{aligned} w &= \langle\langle i_0, \ell_0, r_0\rangle \ldots \langle i_k, \ell_k, r_k\rangle\rangle \\ &= \prod_{a=0}^{k} p_a^{1+2^{i_a+1}.3^{\ell_a+1}.5^{r_a+1}} \end{aligned}$$

where $k \leq F_i(n+1)$.

Since M can only move k squares in k steps, ℓ_a and r_a are bounded by 2^{k+n+1}.

Let M have s states. Then $w \leq g(n)$, where g is the primitive recursive function

$$g(n) = \prod_{a=0}^{k} p_a^{1+2^{s+1}.3^{2^{F_i(n+1)+n+1}+1}.5^{2^{F_i(n+1)+n+1}+1}}.$$

So

$$f(x) = \mathit{Result}((\mu w \leq g(n))[T_e(x, w)]). \ \Box$$

By this result, it seems hard to imagine a useful function that is recursive but not primitive recursive. We would like to classify a set of "feasible" functions such that

$$\begin{aligned} \text{feasible functions} &\subset \text{primitive recursive functions} \\ &\subset \text{recursive functions} \\ &\subset \text{all functions} \end{aligned}$$

What does it mean for a function f to be feasible? Answer: on reasonable inputs x (i.e. where the answer is interesting), it is not totally out of the question to find $f(x)$. Polynomial time is the common model for feasibility.

Definition: Assume that M has alphabet Σ with $\{\flat, 0, 1\} \subseteq \Sigma$. M is a polynomial time Turing machine if there exists a polynomial $p(n)$ such that for all $x \in (\Sigma \setminus \{\flat\})^*$, M, on input x, has run-time $\leq p(n)$, where n is the number of symbols in x (i.e. $n = |x|$).

Definition: A function $f : N \to N$ is polynomial time computable if and only if there exists a Turing machine, M, such that:

- M is a polynomial time Turing machine with alphabet Σ and $\{\flat, 0, 1\} \subseteq \Sigma$, and

- for all $x \geq 0$, M, on input x in binary notation, outputs $f(x)$ in binary notation.

Examples

- $f : x \mapsto 2x$ can be computed in $2n + 3$ steps. It is important to note that we want the computation to be polynomially bounded by n and not x. For example, if $x = 10^9$, then M takes $2\log_2 10^9 + 3 \approx 63$ steps. However if M was bounded by x, then M would take $2 \cdot 10^9 + 3$ steps.
- Factoring integers: $f(x) = \mu p[\mathrm{Pr}\, p \ \wedge \ (x > 1 \rightarrow p|x)]$. The naive method is:

```
for $i = 2, \sqrt{x}$
    if $i|x$ then
        output $i$
        halt
end
output $\sqrt{x}$
```

  This method could loop up to $\sqrt{x}$ times. Finding $\sqrt{x}$ is feasible since we can do a binary search of $1 \ldots x$; this takes $\log_2 x$ iterations which is just n, the number of symbols of x in binary. But looping $\sqrt{x}$ times in terms of n is $2^{n/2}$ times which is not polynomial. So this algorithm for factoring integers is not feasible.

Effective Enumeration

Math 260A - Mathematical Logic

October 24, 1988

(From Last Time)

Last time we said that polynomial time computation was feasible. Why is polynomial time good? What if the polynomial is 10^{100}? - That is certainly not feasible. There are two major reasons why polynomial computation is considered feasible:

- Most natural problems which have polynomial time algorithms turn out to be feasible; e.g. the degree of the polynomial is 3 or 4 at most.
- Polynomial time computation is invariant of the model of computation. For example a $f(n)$ time algorithm for a Random Access Machine (RAM) translates to an $O(f(n)^2)$ time algorithm on a Turing machine. (Note that this is not true for linear time algorithms.)

Another proposal for feasibility is $O(n \log^{O(1)} n)$ time on a RAM as compared with $O(n^{O(1)})$ time. ($O(n^{O(1)}) = O(n^k)$ for some k, and similarly for $O(n \log^{O(1)} n)$).

Definition: P is the class of predicates $A \subseteq N$ (or $A \subseteq \Sigma^*$ for finite Σ) such that χ_A is polynomial time computable. I.e., P is the class of feasible predicates.

Effectively Enumerable

Definition: A set $A \subseteq N$ (or $A \subseteq \Sigma^*$ for finite Σ) is effectively enumerable if and only if there is an effective procedure which lists every member of A. Finite sets and the empty set are effectively enumerable.

In listing A, we may or may not allow repetitions. (But we get the same result either way.) Suppose we have an effective algorithm, M, that lists

$a_1, a_2, \ldots$. Then another effective algorithm, N, would list $a_1, a_{i_1}, a_{i_2}, \ldots$ where a_{i_k} is the $(k+1)^{st}$ element to appear in $a_1, a_2, \ldots$. N runs M, saving everything M outputs. Whenever M outputs a_i, N checks if a_i has already been output; if not, then N outputs it.

Unofficial Equivalent Definition: A is effectively enumerable if and only if A is empty or A is the range of an effectively computable total function. i.e. $n \mapsto a_n$. ("Total" could be left out of the definition, and the definition would still hold.)

Theorem: The set of indices of Turing machines that halt on the empty tape is effectively enumerable. (Recall that this is not decidable.)

Proof: The following algorithm enumerates Turing machines that halt on the empty tape:

```
for $i = 0, 1, \ldots$
    for $j = 0, 1, \ldots, i$
        run Turing machine $j$ on the empty tape for $i$ steps
        if Turing machine $j$ halts then
          output $j$
    end
end
```

Definition: A set is decidable if there is an effective algorithm which can decide whether or not elements are in the set.

Theorem: Every decidable set is effectively enumerable.

Proof: The following algorithm enumerates a decidable set:

```
for $i = 0, 1, \ldots$
    if $i$ is in the set then
      output $i$
end
```

Fact: If A and $N \setminus A$ are both effectively enumerable, then A is decidable.

Definition: A set $A \subseteq N$ is recursively enumerable if and only if A is the domain of some partial recursive function; i.e.

$$A = \{x : f(x) \downarrow\}$$

for some partial recursive function f.

Theorem: Recursively enumerable ⇔ effectively enumerable.

Proof: (via Church's Thesis)

⇐ Suppose A is effectively enumerable. Then we need to find a partial recursive function whose domain is precisely A. Let $f(x)$ be computed as follows:

- Enumerate members of A.
- If and when x appears, output 1 and halt.

⇒ Suppose A is the domain of a partial recursive function f. Then there is a Turing machine, M, which computes f. The following algorithm enumerates A:

```
for $i = 0, 1, \ldots$
    for $j = 1, 2, \ldots, i$
        run $M$ on input $j$ for $i$ steps
         if $M$ on $j$ halts within $i$ steps then
           output $j$
    end
end
```

Propositional Logic

Math 260A - Mathematical Logic

October 26, 1988

The goal that Aristotle set for logic was to “mathematically model intelligent thought”. Not much work was done on this until the late 1800’s when formalisms were developed.

Propositions are the units of propositional logic. They are things which are either true or false; e.g. “gold is denser than water at STP”. Propositions are used to form statements with connectives like $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$. If A and B are propositions, then the meanings of these connectives are:

- $\neg A$ has the opposite truth value that A has.
- $A \wedge B$ is true if and only if A and B are both true.
- $A \vee B$ is true if and only if A or B is true (or both). This is inclusive or as opposed to the exlcusive or usually implied in English.
- $A \rightarrow B$ is true if and only if $\neg A \vee B$ is true. This is “material implication” as opposed to “logical implication”. $\supset$ is sometimes used in place of $\rightarrow$.
- $A \leftrightarrow B$ is true if and only if A and B have the same truth value. $\equiv$ is sometimes used in place of $\leftrightarrow$.

Note that the truth value of a compound proposition depends only on the truth values of its parts.

Propositional Formulas

Propositional formulas are words in the alphabet containing three kinds of symbols:

1)	$P_1, P_2, \ldots$	propositional variables
2)	$\neg, \vee, \wedge, \rightarrow, \leftrightarrow$	propositional connectives
3)	(,)	parentheses

Definition: The propositional formulas are inductively defined by

1. P_i is a propositional formula for all $i \geq 1$.
2. If A and B are propositional formulas, then $(\neg A)$, $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$, and $(A \leftrightarrow B)$ are propositional formulas.
3. Every propositional formula can be obtained by a finite application of (1) and (2).

Note that (P_i) is not a propositional formula.

Equivalent Definition #1: The class of propositional formulas is the smallest class which:

1. contains P_i for all $i \geq 1$.
2. If it contains A and B, then it contains $(\neg A)$, $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$, and $(A \leftrightarrow B)$.

Equivalent Definition #2: The class of propositional formulas is the intersection of all classes satisfying (1) and (2) in *Equivalent Definition #2* above.

Omitting Parentheses

In "colloquial mathematics" we have conventions for omitting parentheses to make formulas more readable.

- $\neg$ has the highest precedence; i.e. it applies to the "smallest amount possible". For example, $\neg A \vee B$ means $((\neg A) \vee B)$, not $(\neg(A \vee B))$.
- $\wedge$ and $\vee$ have the second highest precedence, but they are not to be mixed. $A \vee B \vee C$ means $(A \vee (B \vee C))$; i.e. these connectives are right-associative. $A \vee B \wedge C$ is not a valid name for a formula.
- $\rightarrow$ and $\leftrightarrow$ have the lowest precedence. They are also right-associative, and they are not to be mixed. Be careful about $\rightarrow$; $A \rightarrow (B \rightarrow C)$ is not the same as $(A \rightarrow B) \rightarrow C$.

Truth Tables

The following example is a truth table showing that, unlike $\rightarrow$, $(A \leftrightarrow B) \leftrightarrow C$ *is* the same as $(A \leftrightarrow B) \leftrightarrow C$.

A	B	C	$(A \leftrightarrow B)$	$(B \leftrightarrow C)$	$(A \leftrightarrow B) \leftrightarrow C$	$A \leftrightarrow (B \leftrightarrow C)$
T	T	T	T	T	T	T
T	T	F	T	F	F	F
T	F	T	F	F	F	F
T	F	F	F	T	T	T
F	T	T	F	T	F	F
F	T	F	F	F	T	T
F	F	T	T	F	T	T
F	F	F	T	F	F	F

Note that the fifth and sixth columns are the same.

Definition: A truth assignment is a mapping, $\sigma : \{P_1, P_2, \ldots\} \rightarrow \{T, F\}$, that maps to each propositional variable a value of true or false.

Definition: A truth assignment σ is *extended* to a mapping $\bar{\sigma}$ from propositional formulas to truth values by the following inductive definition (note that extended means that $\bar{\sigma}$ agrees with σ on the propositional variables):

$$\begin{aligned}
\bar{\sigma}(P_i) &= \sigma(P_i) \text{ for all } i \\
\bar{\sigma}(\neg A) &= \begin{cases} T & \text{if } \bar{\sigma}(A) = F \\ F & \text{if } \bar{\sigma}(A) = T \end{cases} \\
\bar{\sigma}(A \wedge B) &= \begin{cases} T & \text{if } \bar{\sigma}(A) = T \text{ and } \bar{\sigma}(B) = T \\ F & \text{otherwise} \end{cases} \\
\bar{\sigma}(A \vee B) &= \begin{cases} T & \text{if } \bar{\sigma}(A) = T \text{ or } \bar{\sigma}(B) = T \\ F & \text{otherwise} \end{cases} \\
\bar{\sigma}(A \rightarrow B) &= \begin{cases} T & \text{if } \bar{\sigma}(A) = F \text{ or } \bar{\sigma}(B) = T \\ F & \text{otherwise} \end{cases} \\
\bar{\sigma}(A \leftrightarrow B) &= \begin{cases} T & \text{if } \bar{\sigma}(A) = \bar{\sigma}(B) \\ F & \text{otherwise} \end{cases}
\end{aligned}$$

The key thing to note here is that the truth value of a propositional formula depends only on the truth values of its parts.

Syntax vs. Semantics of Propositional Logic

Math 260A - Mathematical Logic

October 28, 1988

Syntax of Propositional Logic

Recall that propositional formulas were defined inductively. The following is an example of a proof by induction on the complexity of formulas.

Proposition: The number of occurrences of propositional variables in a formula is equal to 1 plus the number of binary connectives in the formula.

Proof: by induction on the complexity of formulas.

Basis: True for all formulas P_i.

Induction:

Case 1. A is $(\neg B)$. By the inductive hypothesis, the proposition holds for B. The number of occurrences of variables in A is equal to the number of occurrences of variables in B, and the number of binary connectives in A is equal to the number of binary connectives in B. So the proposition holds for A.

Case 2. A is $(B * C)$ for $* \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$. The number of occurrences of variables in A is equal to the number of occurrences of variables in B plus the number of occurrences of variables in C. The number of binary connectives in A is equal to 1 plus the number of binary connectives in B plus the number of binary connectives in C. So, by the inductive hypothesis for B and C, the number of occurrences of variables in A is equal to 1 plus the number of binary connectives in A. □

Theorem: If $A = a_1 a_2 \ldots a_k$ with $a_i \in \{\wedge, \vee, \neg, \rightarrow, \leftrightarrow, (,), P_\ell\}$, then for all $1 \leq j < k$, the number of)'s in $a_1 \ldots a_j <$ the number of)'s in $a_1 \ldots a_j$.

Proof: Homework.

Unique Readability Lemma (only one way to parse a formula): If A is a propositional formula and its i^{th} symbol is a (, then there is a unique $j > i$ such that the i^{th} through j^{th} symbols of A form a propositional formula.

Equivalently, if two subformulas of A overlap, then one is a subformula of the other. (Note that in general, a subformula doesn't have to be proper.)

In order to decide the set of propositional formulas, we need to limit them to a finite alphabet. We can code them using the alphabet $\{\vee, \wedge, \neg, \rightarrow, \leftrightarrow, (,), P, 0, 1, \ldots, 9\}$. E.g. $\neg P_2 \vee P_3$ is $((\neg P2) \vee P3)$.

Theorem: The set of propositional formulas is decidable. (In fact, it is polynomial time decidable since it is a CFG.)

Semantics

Recall truth assignments and their extension to formulas.

Definition: A formula A is a *tautology* (or is valid) if $\bar{\sigma}(A) = T$ for all truth assignments σ. A is *satisfiable* if and only if there exists a truth assignment σ such that $\bar{\sigma}(A) = T$.

A is satisfiable if and only if $(\neg A)$ is not a tautology. A is unsatisfiable if and only if A is not satisfiable, if and only if $(\neg A)$ is a tautology.

Definition: If Γ is a set of formulas, then Γ is satisfiable if and only if there exists a truth assignment σ such that for all $A \in \Gamma$, $\bar{\sigma}(A) = T$.

To show that a formula is a tautology, we can use the method of truth tables. For example, if F is the formula $(A \rightarrow (B \rightarrow C)) \leftrightarrow ((A \wedge B) \rightarrow C)$, then F can be shown to be a tautology as follows:

A	B	C	$(B \rightarrow C)$	$(A \wedge B)$	$(A \rightarrow (B \rightarrow C))$	$((A \wedge B) \rightarrow C)$	F
T	T	T	T	T	T	T	T
T	T	F	F	T	F	F	T
T	F	T	T	F	T	T	T
T	F	F	T	F	T	T	T
F	T	T	T	F	T	T	T
F	T	F	F	F	T	T	T
F	F	T	T	F	T	T	T
F	F	F	T	F	T	T	T

Since the column labeled F is always true, F is a tautology. An implicit assumption that we made is that we only need to check a finite number of truth assignments; i.e. only the truth assignments that differ in their assignment of values to the variables A, B, and C in F.

Theorem: The set of tautologies is decidable.

Proof: The algorithm can use the method of truth tables. It only needs to check the finite number of partial truth assignments to variables occuring in the propositional formula.

Digression

Open Question: Is the set of tautologies polynomial time recognizable? (Equivalent to $P \overset{?}{=} NP$.)

Facts:

- A is satisfiable if and only if $(\neg A)$ is not a tautology.
- A is a tautology if and only if $(\neg A)$ is not satisfiable.
- The set of satisfiable formulas is NP-complete.
- The set of unsatisfiable formulas is co-NP-complete.
- The set of tautologies is co-NP-complete.

Definition: A set $T \subseteq \Sigma^*$, with Σ a finite alphabet, is in NP if and only if there is a polynomial $p(n)$ and a polynomial binary relation $R(x, y)$ such that membership in T is given by

$$\forall x[x \in T \Leftrightarrow \exists y \in \Sigma^*[|y| \leq p(|x|) \text{ and } R(x, y)]].$$

Theorem: The set of satisfiable formulas is in NP because for all x, x is the code of a satisfiable formula if and only if there exists a partial truth assignment σ such that $\bar{\sigma}(\text{formula coded by } x) = T$, and the $|\text{code of } \sigma| \leq |x|$.

Disjunctive Normal Form

Math 260A - Mathematical Logic

October 31, 1988

Digression

> $(A \vee \neg A)$, the law of excluded middle, is a tautology which intuitionists don't believe in. "Normal" mathematicians, called Platonists, believe that mathematical objects have a real existence in some ideal world; so anything you say about them is either true or false.

Definition: A formula is a *literal* if it is of the form P_i or of the form $(\neg P_i)$.

Definition: A formula A is in *disjunctive normal form* (DNF) if and only if it a disjunction $(\vee)$ of conjunctions $(\wedge)$ of literals. I.e. A is of the form $A_1 \vee A_2 \vee \ldots \vee A_k$ with each A_k of the form $B_{k,1} \wedge B_{k,2} \wedge \ldots \wedge B_{k,i_k}$ where each $B_{i,j}$ is a literal.

Theorem: For any formula A, there is a DNF formula B such that $A \leftrightarrow B$ is a tautology.

Proof: One way to prove this is by induction on the complexity of A. Another way requires some more definitions and another theorem:

> *Definition*: A k-ary boolean function is a mapping from $\{T, F\}^k$ $\rightarrow \{T, F\}$.
>
> *Definition*: If f is a k-ary boolean function, then the formula A_f *represents* f if for all truth assignments σ, $\bar{\sigma}(A_f) = f(\sigma(P_1), \ldots, \sigma(P_k))$. (The P_i's are variables in A.)
>
> *Theorem*: If f is a k-ary boolean function, there there is a formula A_f in DNF which represents f.
>
> *Proof*:

Case 1 f is the constant false function. Take A_f to be $P_1 \wedge \neg P_1$.

Case 2 f is not always false. (The idea is to find every case in which f is true, and or them all together.) Let $\vec{t} \in \{T, F\}^k$. Let $C_{\vec{t}}$ be the formula $D_1 \wedge D_2 \wedge \ldots \wedge D_k$ where D_i is P_i if t_i is true and D_i is $\neg P_i$ if t_i is false. $C_{\vec{t}}$ asserts that $P_1, \ldots, P_k$ have truth values $t_1, \ldots, t_k$. Let $A_f = \bigvee_{f(\vec{t})=T} C_{\vec{t}}$. □

(Note that there are at most 2^k values for $\vec{t}$ such that $f(\vec{t}) = T$.)

(Note also that this may not lead to the simplest way to express a formula in DNF. For example, if $f(t_1, t_2) = t_1$, then the above process yields $A_f = (P_1 \wedge P_2) \vee (P_1 \wedge \neg P_2)$, but $A_f = P_1$ is simpler.)

(Note that the general problem of finding the simplest formula B equivalent to a given formula A is hard. A solution to this problem can be used to determine if A is a tautology.)

(back to the proof of finding a DNF formula B equivalent to a given formula A):

Let A involve the variables $P_1 \ldots P_k$. Define f to be the k-ary boolean function $f(t_1 \ldots t_k) = \bar{\sigma}(A)$ where $\sigma(P_i) = t_i$. Let B be a DNF formula which represents f (the existence of which is guaranteed by the above theorem). Now, $\bar{\sigma}(A) = f(\sigma(P_1), \ldots, \sigma(P_k) = \bar{\sigma}(B)$. □

(And this means that $\bar{\sigma}(A \leftrightarrow B) = T$ for all truth assignments σ.

Theorem: If $B \leftrightarrow C$ is a tautology and if A^* is obtained from A by replacing a sub-formula B in A by C, then $A \leftrightarrow A^*$ is a tautology.

Proof: By induction on the complexity of A.

Notation: $\models A$ means that A is a tautology.

Definition: $\Gamma \models A$ if and only if for all truth assignments σ, if $\bar{\sigma}(B) = T$ for all $B \in \Gamma$, then $\bar{\sigma}(A) = T$. ($\Gamma \models A$ intuitively means that the set of formulas Γ logically implies the formula A.)

Notation: $A \models B$ means that $\{A\} \models B$.

Theorem: $\emptyset \models A$ if and only if $\models A$.

Definition: Γ is *finitely satisfiable* if and only if every finite subset of Γ is satisfiable.

Compactness Theorem: Γ is satisfiable if and only if Γ is finitely satisfiable.
Proof:

$\Rightarrow$ Obvious.

$\Leftarrow$ Next time.

Compactness Theorem

Math 260A - Mathematical Logic

November 2, 1988

Theorem: A set Γ of propositional formulas is satisfiable if and only if it is finitely satisfiable.

Proof:

$\Rightarrow$ Obvious.

$\Leftarrow$ In order to prove this direction, we'll need the following lemma.

> *Lemma*: If Γ is finitely satisfiable and A is a formula, then either $\Gamma \cup \{A\}$ or $\Gamma \cup \{\neg A\}$ is finitely satisfiable.
>
> *Proof*: Suppose that both $\Gamma \cup \{A\}$ and $\Gamma \cup \{\neg A\}$ are not finitely satisfiable. Then there exists a finite set $\Sigma_1 \subseteq \Gamma \cup \{A\}$ which is not satisfiable, and there exists a finite set $\Sigma_2 \subseteq \Gamma \cup \{\neg A\}$ which is not satisfiable. In fact, $\Sigma_1 = \Gamma_1 \cup \{A\}$ and $\Sigma_2 = \Gamma_2 \cup \{\neg A\}$ where Γ_1 and Γ_2 are finite subsets of Γ. Consider $\Gamma_1 \cup \Gamma_2$. This is a finite subset of Γ. Since Γ is finitely satisfiable, there exists a truth assignment σ which makes every formula in $\Gamma_1 \cup \Gamma_2$ true.
>
> **Case 1.** $\bar{\sigma}(A) = T$. Then $\Gamma_1 \cup \Gamma_2 \cup \{A\}$ is satisfiable by σ, and in particular, $\Gamma_1 \cup \{A\}$ is satisfiable.
>
> **Case 2.** $\bar{\sigma}(A) = F$. Then $\Gamma_1 \cup \Gamma_2 \cup \{\neg A\}$ is satisfiable by σ, and in particular, $\Gamma_1 \cup \{\neg A\}$ is satisfiable.
>
> So either Σ_1 or Σ_2 is satisfiable which contradicts our assumption. $\Box$

Next we'll define an infinite sequence of sets of propositional formulas. Let $\Pi_0 = \Gamma$ (remember, in this direction, were assuming that Γ is finitely satisfiable), and let

$$\Pi_{i+1} = \begin{cases} \Pi_i \cup \{P_i\} & \text{if } \Pi_i \cup \{P_i\} \text{ is finitely satisfiable} \\ \Pi_i \cup \{\neg P_i\} & \text{if } \Pi_i \cup \{\neg P_i\} \text{ is finitely satisfiable} \end{cases}$$

By the lemma and induction on i, each Π_i is finitely satisfiable.

Facts:

- $\Pi = \bigcup_{i=0}^{\infty} \Pi_i$ is also finitely satisfiable since any finite subset of Π is a subset of Π_i for large enough i.
- $\Gamma \subseteq \Pi$.
- For all i, either P_i or $(\neg P_i)$ is in Π.

Let σ be the truth assignment such that

$$\sigma(P_i) = \begin{cases} T & \text{if } P_i \in \Pi \\ F & \text{if } \neg P_i \in \Pi. \end{cases}$$

Claim: For all $A \in \Pi$, $\bar{\sigma}(A) = T$.
Proof: Let $A \in \Pi$, and say that A uses only the variables $P_1, \ldots, P_k$. Take

$$\{A\} \cup \{P_i \in \Pi \;:\; i \leq k\} \cup \{(\neg P_i) \in \Pi \;:\; i \leq k\},$$

which is a finite subset of Π, and hence is satisfiable by some truth assignment, τ.
Since τ satisfies this set, $\tau(P_i) = T$ if and only if $P_i \in \Pi$ for $i \leq k$. So $\tau(P_i) = \sigma(P_i)$ for all $i \leq k$. Hence $\bar{\sigma}(A) = \bar{\tau}(A) = T$. □

So σ makes every formula in Π and hence every one in Γ true. So Γ is satisfiable. □

Notation: $\sigma \models \Gamma$ means that σ satisfies Γ; i.e. for all $A \in \Gamma$, $\bar{\sigma}(A) = T$. $\Gamma \models A$ means that A is a logical consequence of Γ. Γ is inconsistent if and only if Γ is unsatisfiable.

Theorem (rephrased): Γ is inconsistent if and only if some finite subset is inconsistent.

Theorem: $\Gamma \models A$ if and only if $\Gamma \cup \{\neg A\}$ is inconsistent.

Proof: For all truth assignments σ such that $\sigma \models \Gamma \rightarrow \bar{\sigma}(A) = T$,

$$\begin{aligned}
\neg\exists\sigma[\sigma \models \Gamma \cup \{\neg A\}] &\equiv \forall\sigma[\sigma \not\models \Gamma \cup \{\neg A\}] \\
&\equiv \forall\sigma[\sigma \not\models \Gamma \text{ or } \bar{\sigma}(\neg A) = F] \\
&\equiv \forall\sigma[\sigma \models \Gamma \rightarrow \bar{\sigma}(\neg A) = F] \\
&\equiv \forall\sigma[\sigma \models \Gamma \rightarrow \bar{\sigma}(A) = T] \;\Box
\end{aligned}$$

Theorem: If $\Gamma \models A$ then there is a finite subset $\Gamma_0 \subseteq \Gamma$ such that $\Gamma_0 \models A$.

Proof: $\Gamma \cup \{\neg A\}$ is inconsistent implies that there is a finite $\Gamma_0 \subseteq \Gamma$ such that $\Gamma_0 \cup \{\neg A\}$ is inconsistent.

Theorem: If Γ is effectively enumerable, (and hence r.e.) then the set of logical consequences of Γ is effectively enumerable.

Proof: Let M be an effective procedure for listing Γ as $\{\gamma_1, \gamma_2, \ldots\}$. Let $A_1, A_2, \ldots$ be an effective enumeration of all formulas. Then the following algorithm enumerates the logical consequences of Γ:

```
for $i$ = $1, 2, \ldots$
    run $M$ to get $\gamma_i$
    for $j$ = $1, 2, \ldots, i$
        if $\{\gamma_1, \gamma_2, \ldots, \gamma_i\} \models A_j$ then
          output $A_j$
    end
end
```

Note the following:

- When M is run to get the next γ_i, we already know $\gamma_1 \ldots \gamma_{i-1}$.
- Checking if $\{\gamma_1, \gamma_2, \ldots, \gamma_i\} \models A_j$ can be done by truth tables.
- If a list without repetitions is desired, a check could be made before outputting each A_j. $\Box$

Corollary: If Γ is finite (or if Γ is decidable), then $\{A : \Gamma \models A\}$ is effectively enumerable.

Enumerating Formulas, 1st Order Logic

Math 260A - Mathematical Logic

November 4, 1988

We can code propositional formulas in the language $\{(,), \neg, \wedge, \vee, \rightarrow, \leftrightarrow, P, 0, \ldots, 9\}$. Then two ways of effectively enumerating propositional formulas as $A_1, A_2, \ldots$ are:

1.
```
for $i = 2, 3, \ldots$
    list out all formulas with exactly $i$ symbols
end
```

2.
```
for $i = 2, 3, \ldots$
    list out all formulas with less than $i$ propositional
        connectives that involve only variables $P_1 \ldots P_i$
end
```

Theorem: Suppose that Γ is effectively enumerable, and suppose also that for every formula A, $\Gamma \models A$ and/or $\Gamma \models \neg A$ (i.e. Γ is complete). Then the set of logical consequences of Γ is decidable.

Proof: Recall that the consequences of Γ can be enumerated. The idea is that given A, we will list out the consequences of Γ until either A or $\neg A$ appears. If A appears, then A is a logical consequence of Γ, and if $\neg A$ appears, then A is not a logical consequence of Γ provided that Γ is not inconsistent (or is consistent). There are two possible cases for Γ:

Case 1. Γ is inconsistent. Then $\Gamma \models A$ for *all* A. So the set of logical consequences of Γ is the set of all formulas which is effectively enumerable.

Case 2. Γ is consistent. Then use the algorithm of the above idea. □

Even though there are only two possibilities for Γ, we don't have any effective way of deciding whether or not any given Γ is consistent. To see this, consider the following procedure for creating Γ_M based on a Turing machine, M:

```
$\Gamma_M = \emptyset$
for $i = 1, 2, \ldots$
    $\Gamma_M = \Gamma_M \cup P_i$
    run $M$ for $i$ steps
    if $M$ halts then
      $\Gamma_M = \Gamma_M \cup \neg P_1$
      halt
    end
end
```

The idea is that for every step of the Turing machine M, we add a new propositional variable to Γ_M. As long as M keeps running, Γ_M remains consistent. As soon as we reach a halting configuration of M, we make Γ_M inconsistent by adding $\neg P_1$ to it. Clearly Γ_M is effectively enumerable. But Γ_M is consistent iff M never halts. So if we could determine whether an arbitrary Γ was consistent or not, we could solve the halting problem.

One way to show that $\Gamma \models \phi$ is to show that $\Gamma \cup \{\neg\phi\}$ is inconsistent. A proof method for showing $\Gamma \models \phi$ is to derive a contradiction from $\Gamma \cup \{\neg\phi\}$. (One way of doing this is to use the method of truth tables.)

First Order Logic

(See also instructor's notes entitled First Order Logic.) First order logic has variables that range over a non-empty set of objects instead of just T and F. It also has functions and predicates that operate on these objects, and quantifiers $\forall$ and $\exists$ which range over the set of objects.

For example, let the domain of objects (the universe) be the set of all people, and let the binary relation "Loves(x, y)" be true if x loves y and false otherwise. Then we can translate the following sentences into first order logic. "Alma loves someone" - $\exists x$ Loves(Alma,x). "Someone loves Alma" - $\exists x$ Loves(x,Alma). "None of Alma's lovers' lovers, love Alma" - $\forall x(\exists x(\text{Loves}(x, y) \wedge \text{Loves}(y, \text{Alma}) \rightarrow \neg\text{Loves}(x, \text{Alma}))$. Note that this implies $\neg$Loves(Alma, Alma).

Another example is "Mother(x)", a unary function. In first order logic, functions will always be single valued and total. So in order to state that someone has no mother, we need a relation “Motherof(x, y)”. “x has no mother” - $\forall y(\neg$Motherof(y, x) or $\neg\exists y($Motherof$(y, x))$. In these formulas, x is a free variable; i.e. it appears as a parameter in the formula. y is bound.

Free and bound variables are treated differently when we combine formulas. The sentence “There are exactly two people with no mother”, translates to $\exists x \exists y(y \neq x \wedge$ “x has no mother” $\wedge$ “y has no mother” $\wedge \neg\exists z(z \neq x \wedge z \neq y \wedge$ “z has no mother”$))$. When expanding “y has no mother”, we have to rename the bound variable y so it won't clash with the y bound by the second $\exists$; i.e. $\neg\exists y^{'}($Motherof$(y^{'}, y))$.

Defining a Logic

Math 260A - Mathematical Logic

November 7, 1988

(Guest lecturer: Jeff Remmel)

To define a logic, you:

1. Define formulas (meaningful sequences, well formed formulas). I.e. $(A \vee B)$ is a formula, but not $A \neg \wedge \vee$.
2. Establish unique readability. I.e. $A \vee B \wedge C$ means either $(A \vee B) \wedge C$ or $A \vee (B \wedge C)$. If A is true and B and C are false, then $A \vee B \wedge C$ is false in the first case and true in the second.
3. Inductively define when a truth assignment (model) satisfies a formula.
4. Develop a proof theory.
5. Tie together satisfaction and proof theory to show completeness.

The language of first order logic contains ... (see page 3 of instructor's notes). For example, let a language be

$\forall$	- is intended to mean for all things
$0, 1$	- constant symbols for zero and 1
$+, -$	- binary functions for addition and subtraction
N	- unary relation intended to mean "is a number"
I	- unary relation intended to mean "is interesting"
P	- unary relation intended to mean "is a person"
T	- unary relation intended to mean "is a time"
$<$	- binary relation for less than
F	- binary relation: Fxy - "you can fool x at y"

Then the following sentences are expressible.

"There are at least three interesting numbers" -

$$\begin{aligned}\exists x_1 \exists x_2 \exists x_3 (I(x_1) \wedge I(x_2) \wedge I(x_3)\\ \wedge N(x_1) \wedge N(x_2) \wedge N(x_3)\\ \wedge x_1 \neq x_2 \wedge x_1 \neq x_3 \wedge x_2 \neq x_3).\end{aligned}$$

(Note that $I(x_i)$ really should be Ix_i, and that $x_i \neq x_j$ really should be $(\neg(= x_i x_j))$. Sometimes we use colloquialisms for better readability.)

"There are exactly three interesting numbers" -

$$\begin{aligned}&\exists x_1 \exists x_2 \exists x_3 (I(x_1) \wedge I(x_2) \wedge I(x_3)\\ &\quad \wedge N(x_1) \wedge N(x_2) \wedge N(x_3)\\ &\quad \wedge x_1 \neq x_2 \wedge x_1 \neq x_3 \wedge x_2 \neq x_3)\\ &\quad \forall x_4 (I(x_4) \wedge N(x_4) \rightarrow x_1 = x_4 \vee x_2 = x_4 \vee x_3 = x_4)).\end{aligned}$$

"You can fool all the people some of the time" - there are two interpretations of this. There is one particular time at which you can fool everyone:

$$\exists x_1 (T(x_1) \wedge \forall x_2 (P(x_2) \rightarrow F(x_2, x_1))),$$

or any person can be fooled at some time:

$$\forall x_2 (P(x_2) \rightarrow \exists x_1 (T(x_1) \wedge F(x_2, x_1))).$$

The commutative law for addition -

$$\forall x \forall y (+(x, y) = +(y, x)).$$

To define meaningful phrases, we need to define terms, formulas, and sentences. (See pages 5-6 of instructor's notes.)

Sentences

Consider the formula "$= x5$". What is 5? Assume it's the natural number five. What is x? We need to quantify it; i.e. $\forall x (= x5)$. What does $\forall$ range over?

In order to answer these question, we introduce the concept of a model. Intuitively, a model gives an interpretation of the non-logical symbols. A model $\mathcal{M}$ consist of

- A universe, M. This is what $\forall$ and $\exists$ range over.
- $\mathcal{M}(c) \in M$ - constants in the universe.
- $\mathcal{M}(R) \subseteq M^k$ - set of k-tuples for which R holds.
- $\mathcal{M}(f) : M^k \to M$ - functions on the universe.

A sentence is a formula with no free variables. (See page 9 of instructor's notes.)

The problem we have is that we want to define the truth of formulas inductively, but we don't want to include formulas like '$= x5$'.

Unique Readability of Terms, Models

Math 260A - Mathematical Logic

November 9, 1988

(Guest lecturer: Jeff Remmel)

(Terms are defined on page 5 of instructor's notes.)

Unique Readability Lemma (alternate statement): No proper initial prefix of a term is a term. If t is a term and $t^{'}$ is a proper initial segment of t, then $t^{'}$ is not a term.

Proof: by induction on the length, n, of t.

Basis: $n = 1$. Then $t = x$ or $t = c$, so $t^{'} = \epsilon$ which is not a term.

Induction: Assume the lemma true for all terms of length $\leq n$. Consider a term of length $n + 1$. $t = f\alpha_1 \ldots \alpha_k$ were f is a k-ary function symbol, and $\alpha_1 \ldots \alpha_k$ are terms.

Suppose that $t^{'} \neq \epsilon$ and that $t^{'}$ is a term. Then $t^{'} = f\beta_1 \ldots \beta_2$, where $\beta_1 \ldots \beta_2$ are terms. Compare α_1 and β_1.

Case 1. $\alpha_1 = \beta_1$.

Case 2. α_1 is an initial segment of β_1. This is not possible by the induction hypothesis.

Case 3. β_1 is an initial segment of α_1. This is not possible by the induction hypothesis.

By applying the argument k times, we get

$$\alpha_1 = \beta_1,\ \alpha_2 = \beta_2,\ \ldots \alpha_k = \beta_k.$$

This implies that $t^{'} = t$ which contradicts our choice of $t^{'}$.

Models

Consider the sentence $\forall x(x \neq 0 \rightarrow \exists y(xy = 1))$. This is true in the rationals Q and the reals R since every non-zero number has an inverse. But it is false in the natural numbers N. So the truth of a sentence in first order logic depends on something else which we call a model. (See page 10 of instructor's notes.)

Our goal is to define truth inductively. The problem is that in order to define the truth of sentences, you have to look at formulas with free variables. (I.e. an inductive definition of truth for the sentence $\forall x(x = x)$ depends on the truth of the formula $x = x$ which has a free variable.)

Truth

Math 260A - Mathematical Logic

November 14, 1988

Truth depends on a structure. The reason we want structures to have non-empty universes is that we want $(\forall x\phi(x) \to \exists x\phi(x))$ to be true. (It is vacuously false in the empty universe.)

Structures

(Note that structures don't include an interpretation for $=$.) Consider the language with

$<$ - a binary relation
0 - a constant symbol
S - a unary function symbol

The standard interpretation of this language is $\mathcal{N} = (N, 0^{\mathcal{N}}, S^{\mathcal{N}}, <^{\mathcal{N}})$ where N is the set of natural numbers, $0^{\mathcal{N}}$ is the natural number zero, $S^{\mathcal{N}}$ is the successor function ($S^{\mathcal{N}} : N \to N$, $S^{\mathcal{N}}(x) = x + 1$), and $<^{\mathcal{N}}$ is the less than relation ($<^{\mathcal{N}} \subseteq N \times N$, $a <^{\mathcal{N}} b$ iff a is less than b).

Since the following sentences are all true in $\mathcal{N}$, the structure is said to satisfy them:

1. $\forall x(Sx \neq 0)$
2. $\forall x\forall y(\neg(x < y \wedge y < Sx))$
3. $\forall x(x < Sx)$
4. $\forall x(x = 0 \vee 0 < x)$
5. $\forall x(x \neq 0 \to \exists y(Sy = x)$

Another structure which also satisfies the above sentences is $\mathcal{M}$ which is defined as follows. Let the universe, M, be $N \cup Z'$, where Z' is the set of of integers with primes on them; i.e. pictorially, M is

$$0, 1, 2, 3, \ldots \;\; \ldots, -2^{'}, -1^{'}, 0^{'}, 1^{'}, 2^{'}, \ldots$$

$Z^{'}$ represents a set of non-standard objects. (The primes are intended to represent the fact that $N \cap Z^{'} = \emptyset$.)

$\mathcal{M}$ interprets the non-logical language symbols as follows:

$0^{\mathcal{M}} = 0$
$S^{\mathcal{M}}(n) = n + 1$
$S^{\mathcal{M}}(n^{'}) = (n^{'} + 1)^{'}$
$<^{\mathcal{M}} (a, b) \Leftrightarrow a < b$
$<^{\mathcal{M}} (a^{'}, b^{'}) \Leftrightarrow a < b$
$<^{\mathcal{M}} (a, b^{'})$ is true
$<^{\mathcal{M}} (a^{'}, b)$ is false

Another structure which doesn't satisfy all of the above sentences is $\mathcal{R} = (R, 0^{\mathcal{R}}, S^{\mathcal{R}}, <^{\mathcal{R}})$, where R is the set of real numbers, and

$0^{\mathcal{R}}$ is zero
$S^{\mathcal{R}}(x) = x - 1$
$<^{\mathcal{R}} (x, y) \Leftrightarrow x < y$

Sentences 1, 3, and 4 are false in $\mathcal{R}$, while sentences 2 and 5 are true.

Object Assignments, Free and Bound Variables

Math 260A - Mathematical Logic

November 16, 1988

Object assignments assign objects in the universe of a structure to variables. (See pages 12-14 of instructor's notes.)
Examples of truth:

1.

$$\begin{aligned}
&\mathcal{M} \models \forall x_1(x_1 = x_1)[s] \\
&\quad \text{iff} \quad \mathcal{M} \models \forall x_1(x_1 = x_1)[s(a/x_1)] \text{ for all } a \in M \\
&\quad \text{iff} \quad s(a/x_1)(x_1) = s(a/x_1)(x_1) \text{ for all } a \in M \\
&\quad \text{iff} \quad a = a \text{ for all } a \in M
\end{aligned}$$

2.

$$\begin{aligned}
&\mathcal{M} \models \forall x_1(x_1 = x_2)[s] \\
&\quad \text{iff} \quad a = s(x_2) \text{ for all } a \in M \\
&\quad \text{iff} \quad M \text{ has only one object}
\end{aligned}$$

Alternate Definition (of a free variable): (See page 9 of instructor's notes for original definition.) An occurrence of x is free in A iff there is no subformula of the form

$$\forall x \ldots \underline{x} \ldots \text{ or } \exists x \ldots \underline{x} \ldots$$

where $\underline{x}$ is the occurrence of x.

For example, in the formula

$$x = 0 \land \exists x(x \neq 0),$$

the first occurrence of x is free, while the others are bound.

Given an occurrence of x in A, x is bound by an occurrence of quantifier Qx in A (i.e. is in the scope of Q) iff the subformula QxB in A which starts with Qx contains the occurrence of x and x is free in B. For example, in the formula

$$\exists x(x = 0 \wedge \exists x(x \neq 0)),$$

the second occurrence of x is bound by the first occurrence of $\exists x$, while the fourth occurrence of x is bound by the second occurrence of $\exists x$.

Theorem: If A is a formula, then $\mathcal{M} \models A[s]$ does not depend on values $s(y)$ for y not occurring free in A.

Proof: by induction on the complexity of A.

Theorem: Let A be a formula, z be a variable not occurring in A, and Qx be an occurrence of a quantifier in A. Let $A^{'}$ be obtained by changing Qx to Qz and every x bound by Qx in A to z. Then $A^{'}$ is a formula, and $\mathcal{M} \models A[s]$ iff $\mathcal{M} \models A^{'}[s]$ for all $\mathcal{M}$ and s.

Proof: by induction on the complexity of A.

For example, consider the following three formulas:

$$\begin{array}{rcl} A & = & \forall x \forall y(x = y) \\ A^{'} & = & \forall y \forall y(y = y) \\ A^{''} & = & \forall z \forall y(z = y) \end{array}$$

A is equivalent to $A^{''}$ but not $A^{'}$.

Corollary: If A is a sentence, $\mathcal{M} \models A[s]$ is independent of s. (See page 15 of instructor's notes.)

For example, consider the language with $0, S, +, \cdot$, and $\uparrow$ (exponentiation), and the standard model $\mathcal{N}$ containing the set of natural numbers and the usual operations. Fermat's last theorem (FLT) is expressed as:

$$\forall x \forall y \forall z \forall n(n > 2 \wedge x \cdot y \cdot z > 1 \rightarrow x \uparrow n + y \uparrow n \neq z \uparrow n).$$

Now, $\mathcal{N} \models$ FLT iff FLT is true.

As another example, consider the axiom of choice. This is a first order property in the language of set theory. But we don't have a definition of what it means for the axiom of choice to be true because we don't have a standard interpretation for set theory. (Because we don't have a set of all sets.)

Satifiability, Validity, Substitutions, Prenex Normal Form

Math 260A - Mathematical Logic

November 18, 1988

Theorem: Suppose A is a formula with free variables $x_1, \ldots, x_k$. Then $\forall x_1 \ldots \forall x_k A$ is a sentence, and $\models A$ iff $\models \forall x_1 \ldots \forall x_k A$.

Definition: Given a first order formula A, A is a *first order tautology* iff there is a propositional tautology B, with propositional variables $P_1, \ldots, P_k$ and $A = B(A_1/P_1, \ldots, A_k/P_k)$ (i.e. each P_i in B is changed to a first order formula A_i).

For example,

$$\forall x(x = 0) \to \forall x(x = 0)$$

is a tautology because it is of the form $P_1 \to P_1$. An alternate way of looking at the definition is to let the maximal subformulas of the form $Qx(\ldots)$ or that are atomic in A be "propositional variables". Then A should be a tautology.

Theorem: A first order tautology is valid.

Definition: A *tautologically implies* B means that $A \to B$ is a tautology. Γ *tautologically implies* B means that $\forall A \in \Gamma$, $A \to B$ is a tautology.

Definition: A is *logically equivalent* to B ($A \simeq B$) iff $\models A \leftrightarrow B$.

Definition: Γ is *satisfiable* iff there is a structure $\mathcal{M}$ and an object assignment s such that $\mathcal{M} \models \Gamma[s]$. (If Γ consists of just sentences, then the object assignment is not needed.)

For example, consider the even integers; x is even iff there is a y such that $y+y=x$. In $\mathcal{N}$, the standard interpretation of $0, S, +$, and $\cdot$, the even integers are definable as:

$$\{m : \mathcal{N} \models \exists y(y+y=x)[s(m/x)]\}.$$

So the even integers are said to be *first order definable* in $\mathcal{N}$. (Note that the part to the right of the ':' is independent of s since we change the assignment of x.)

Often, we are lazy and write

$$\{m : \mathcal{N} \models \exists y(y+y=m)\}.$$

$\exists y(y+y=m)$ is not a first order formula because m is an integer, not a symbol for an element of the universe.

Theorem: $\Gamma \models A$ iff $\Gamma \cup \{\neg A\}$ is unsatisfiable. (This is the law of the excluded middle.)

Proof: immediate from the definitions.

Substitution

Intuitively, if A says something about x, and we substitute a term t for x, then $A(t/x)$ says the same thing about t.

Definition: Let A be a formula, x be a variable, and t be a term. The formula $A(t/x)$ is obtained as follows:

1. For each variable y occurring in t, if y occurs bound in A (i.e. Qy occurs in A), then pick a new variable z which does not occur in A or t, and change every bound occurrence of y in A to z and change each Qy in A to Qz.

2. Change every free occurrence of x in A to t.

For example, if $A(x)$ is $\exists y(y+y=x)$, then $A(y+z/x)$ is not $\exists y(y+y=y+z)$. Instead, it is $\exists v(v+v=y+z)$, or $\exists w(w+w=y+z)$, or ...

As another example, if $A(x)$ is $\exists y(y+y=x)$, then $A(2\cdot x/x)$ is $\exists y(y+y=2\cdot x)$, not $\exists y(y+y=\cdots 2\cdot 2\cdot x)$

For simultaneous substition, $A(t_1/x_1, \ldots, t_k/x_k)$, do the following

- Rename bound variables in A to avoid variables of $t_1, \ldots, t_k$.

- Replace simultaneously each free occurrence of x_i with t_i.

Note that $A(t_1/x_1, t_2/x_2)$ may not equal $A(t_1/x_1)(t_2/x_2)$ (if t_1 has an occurrence of x_2). Also, note that $A(t/x)$ is ambiguously defined since it allows a choice of new bound variables z; but any two incarnations of $A(t/x)$ are logically equivalent. Finally, note that out of laziness we often write $A = A(x)$ and $A(t)$ for $A(t/x)$.

Prenex Normal Form

A formula is in prenex normal form if all quantifiers are "pulled out" to the front of the formula. For example, $(\exists x Px \to Px)$ is logically equivalent to $\forall y(Py \to Px)$.

Proof:

$$\begin{aligned}
&\mathcal{M} \models (\exists x Px \to Px)[s] \\
&\quad \text{iff} \quad \text{if there is an } a \in M \text{ such that} \\
&\qquad\qquad \mathcal{M} \models Px[s(a/x)],\ \text{then } \mathcal{M} \models Px[s] \\
&\quad \text{iff} \quad \text{if there is an } a \in M \text{ such that} \\
&\qquad\qquad a \in P^M,\ \text{then } s(x) \in P^M \\
&\quad \text{iff} \quad \text{either there is no } a \in M \text{ such that} \\
&\qquad\qquad a \in P^M \text{ or } s(x) \in P^M \\
&\quad \text{iff} \quad \text{either for every } a \in M, \\
&\qquad\qquad a \notin P^M \text{ or } s(x) \in P^M \\
&\quad \text{iff} \quad \text{for every } a \in M,\ \text{either} \\
&\qquad\qquad a \notin P^M \text{ or } s(x) \in P^M \\
&\quad \text{iff} \quad \mathcal{M} \models (\forall y Py \to Px)[s].\ \Box
\end{aligned}$$

Prenex Normal Form Theorem, Isomorphisms

Math 260A - Mathematical Logic

November 21, 1988

Logical Equivalences

If A and B are formulas, and x is not free in B, then the following logical equivalences hold:

$$\begin{array}{rcl} QxA \wedge B & \simeq & Qx(A \wedge B) \\ QxA \vee B & \simeq & Qx(A \vee B) \\ QxA \rightarrow B & \simeq & Q^{'}x(A \rightarrow B) \\ B \rightarrow QxA & \simeq & Qx(B \rightarrow A) \end{array}$$

where $Q^{'}$ is $\forall$ if Q is $\exists$, or $\exists$ if Q is $\forall$.

As an example, consider $\forall xA \rightarrow B \simeq \exists x(A \rightarrow B)$. There are two ways we can show this: by proof or by derivation.

Proof. Fix $\mathcal{M}$ and s. Then there are two cases to consider:

Case 1. $\forall xA$ is true.

(forward) Suppose that $\forall xA \rightarrow B$ is true. Then B is true. So $\exists x(A \rightarrow B)$ is true since there is an $a \in M$ such that $\mathcal{M} \models A \rightarrow B[s(a/x)]$.

(backward) Now suppose that $\exists x(A \rightarrow B)$ is true. Since $\forall xA$ is true, B is true. So $\forall xA \rightarrow B$ is true.

Case 2. $\forall xA$ is false. Then $\mathcal{M} \models \exists x \neg A$. So there is an $a \in M$ such that $\mathcal{M} \models \neg A(a/x)$. Now $\mathcal{M} \models \forall xA \rightarrow B$, so $\mathcal{M} \models A(a/x) \rightarrow B$, and hence $\mathcal{M} \models \exists x(A \rightarrow B)$,

Derivation.

$$\begin{aligned}\forall xA \to B &\simeq \neg(\forall xA) \vee B\\ &\simeq (\exists x(\neg A)) \vee B\\ &\simeq \exists x(\neg A \vee B)\\ &\simeq \exists x(A \to B)\end{aligned}$$

More Logical Equivalences

$$\begin{aligned}\neg\forall xA &\simeq \exists x(\neg A)\\[1em] \neg\exists xA &\simeq \forall x(\neg A)\\[1em] \forall xA \leftrightarrow B &\simeq (\forall xA \to B) \wedge (B \to \forall xA)\\ &\simeq \exists x(A \to B) \wedge \forall x(B \to A)\\ &\simeq \exists y(A(y/x) \to B) \wedge \forall x(B \to A)\\ &\simeq \exists y\forall x(A(y/x) \to B) \wedge (B \to A)\end{aligned}$$

Prenex Normal Form Theorem

Theorem: Every formula is logically equivalent to a prenex normal form formula.

Proof: (outline) The above logical equivalences combined with renaming of bound variables allow us to "pull out" quantifiers to the front of a formula one at a time. (The details involve induction on the complexity of formulas.)

The process of converting a formula to prenex normal form is not unique. For example, consider $\exists xP(x) \to \exists xP(x)$. One conversion produces:

$$\begin{aligned}\exists xP(x) \to \exists xP(x) &\simeq \exists x(\exists xP(x) \to P(x))\\ &\simeq \exists x(\exists yP(y) \to P(x))\\ &\simeq \exists x\forall y(P(y) \to P(x))\end{aligned}$$

and another produces:

$$\begin{aligned}\exists x P(x) \rightarrow \exists x P(x) &\simeq \exists y P(y) \rightarrow \exists x P(x)\\ &\simeq \forall y (P(y) \rightarrow \exists x P(x))\\ &\simeq \forall y \exists x (P(y) \rightarrow P(x))\end{aligned}$$

But any two prenex normal form formulas that are logically equivalent to the same formula are logically equivalent to each other.

As another example of this,

$$\exists x \forall y (P(y) \rightarrow Q(x)) \simeq \forall y \exists x (P(y) \rightarrow Q(x))$$

since these are both prenex normal form formulas for $\exists y P(y) \rightarrow \exists x Q(x)$.

Note that in general, you can't exchange the order of quantifiers in a formula. You can only do this if the x's are on one side and the y's on the other. For example, in expressing continuity,

$$\forall x \forall \epsilon \exists \delta \forall y (\epsilon > 0 \wedge \delta > 0 \wedge |y - x| < \delta \rightarrow |f(x) - f(y)| < \epsilon),$$

you can't change the order of the quantifiers. Similarly for uniform continuity,

$$\forall \epsilon \exists \delta \forall x \forall y (\epsilon > 0 \wedge \delta > 0 \wedge |y - x| < \delta \rightarrow |f(x) - f(y)| < \epsilon).$$

Isomorphisms

Definition: Let L be a language, and $\mathcal{M}$ and $\mathcal{N}$ be structures for L. Then $h : \mathcal{M} \rightarrow \mathcal{N}$ is an isomorphism iff:

- $h : M \rightarrow N$ is 1-1 and onto.
- for every constant symbol $c \in L$ $h(c^{\mathcal{M}}) = c^{\mathcal{N}}$.
- for every function symbol $f \in L$ and every $\langle m_1, \ldots, m_k \rangle \in M$, $h(f^{\mathcal{M}}(m_1, \ldots, m_k) = f^{\mathcal{N}}(h(m_1), \ldots, h(m_k))$.
- for every relation symbol $R \in L$ and every $\langle m_1, \ldots, m_k \rangle \in M$, $R^{\mathcal{M}}(m_1, \ldots, m_k) \Leftrightarrow R^{\mathcal{N}}(h(m_1), \ldots, h(m_k))$. (Note that this automatically holds for '=' since h is 1-1.)

Isomorphism Theorem, Definability

Math 260A - Mathematical Logic

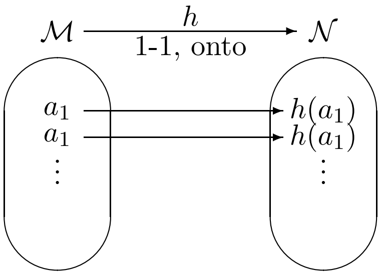
November 23, 1988

Isomorphism Theorem

Theorem: If $\mathcal{M}$ and $\mathcal{N}$ are structures for a common language L, $h : \mathcal{M} \to \mathcal{N}$ is an isomorphism, ϕ is a formula, and s is an object assignment for $\mathcal{M}$, then

$$\mathcal{M} \models \phi[s] \text{ iff } \mathcal{N} \models \phi[h \circ s].$$

Pictorially, if we have



and we have a formula $\phi(x_1, \ldots, x_k)$ with $s(x_1) = a_1, \ldots, s(x_k) = a_k$, then

$$\mathcal{M} \models \phi(a_1, \ldots a_k) \text{ iff } \mathcal{N} \models \phi(h(a_1), \ldots h(a_k)),$$

or

$$\mathcal{M} \models \phi[s] \text{ iff } \mathcal{N} \models \phi[h \circ s].$$

Proof: (outline) First, show that for every term t, $h(\bar{s}(t)) = \overline{h \circ s}(t)$. Show this by induction on t; i.e. show it for constant symbols, and then show that it respects function symbols. Second, prove the theorem by induction on the complexity of (number of logical connectives in) ϕ.

First Order Definability

Definition: Let $\mathcal{M}$ be a structure. A subset, R of M is (first order) definable in $\mathcal{M}$ iff there is a formula $\phi(x)$ (containing only one free variable, x) such that $R = \{m \in M : \mathcal{M} \models \phi(m)\}$. (Note that R may have more than one such form; i.e. $\phi \wedge \phi \wedge \ldots$)

$S \subseteq M^k$ is definable iff there is a formula $\phi(x_1, \ldots, x_k)$ (containing only k free variables) such that $S = \{\langle m_1, \ldots, m_k\rangle : \mathcal{M} \models \phi(m_1, \ldots, m_k)\}$.

A k-ary function $f : M^k \to M$ is definable iff its graph is definable. (The graph of f is $\{\langle m_1, \ldots, m_k, m_{k+1}\rangle : f(m_1, \ldots, m_k) = m_{k+1}\}$.)

Some examples: If G is a group in the language $(0, +, -)$, then the set of elements of order 2 is definable by $x + x = 0$. On the other hand, there is no uniform (i.e. works for every group) formula that defines the objects of finite order. (x is of finite order iff

$$\underbrace{x + x \ldots + x}_{n} = 0$$

for some integer n.) There are groups for which the set of objects of finite order is not first order definable; e.g. $Z_2 \oplus Z_3 \oplus Z_4 \ldots$. If G is finite, then $x = x$ defines the sets of objects of finite order since every object is of finite order.

As an example of something that is not definable, consider the language $L = (0, +, <)$, and the structure $\mathcal{R} = (R, 0, +, <)$.

Claim: In $\mathcal{R}$, multiplication is not definable. I.e. there is no $\phi(x_1, x_2, x_3)$ such that for all $a, b, c \in R$, $\mathcal{R} \models \phi(a, b, c)$ iff $ab = c$.

Proof: Idea: find an automorphism $h : \mathcal{R} \to \mathcal{R}$ which does not preserve multiplication. Let $h : x \mapsto 2x$. Since

$h : \mathcal{R} \to \mathcal{R}$,
h is 1-1,
h is onto,
$x < y \Leftrightarrow h(x) < h(y)$,
$x + y = z \Leftrightarrow h(x) + h(y) = h(z)$, and
$h(0) = 0$,

h is an isomorphism. Suppose, for the sake of contradiction, that $\phi(x_1, x_2, x_3)$ defines (the graph of) multiplication. Then, for all $a, b, c \in R$,

$$\begin{aligned}\mathcal{R} \models \phi(a,b,c) &\Leftrightarrow \mathcal{R} \models \phi(h(a),h(b),h(c)) \\ &\Leftrightarrow \mathcal{R} \models \phi(2a,2b,2c)\end{aligned}$$

or

$$ab = c \Leftrightarrow (2a)(2b) = (2c)$$

which is false for nonzero a, b, and c. □
(Note that in $\mathcal{R}$, $\{1\}$ is not definable by the same automorphism.)

General Principle: Any definable relation or function (in a given structure) is preserved under automorphism. (But not conversely. As an example, $\mathcal{N} = (N, 0, S)$ has no nontrivial automorphism.[1] So every $A \subseteq N$ is preserved under automorphism. But there are uncountably many $A \subseteq N$ and only countably many formulas. So some A are not definable.)

An example of something that is definable in $\mathcal{R}$ is additive inverse; i.e. $f(x) = -x$. The graph of f is $\{\langle x, y\rangle : x + y = 0\}$, so f is definable as

$$\{\langle a, b\rangle : \mathcal{R} \models x + y = 0[s(a/x)(b/y)]\}.$$

[1] A trivial automorphism is the identity automorphism. To show that $\mathcal{N}$ has no nontrivial automorphism, use induction on the size of N.

Classes of Models, Q, Peano Arithmetic

Math 260A - Mathematical Logic

November 28, 1988

Definition: Let Γ be a set of sentences in a language L. $Mod(\Gamma)$ is the class of models (structures) of Γ.

Definition: A class $\mathcal{K}$ of structures for a language L is an *elementary class* (EC) iff there is a finite set Γ of sentences such that $\mathcal{K} = Mod(\Gamma)$. Equivalently, iff there is a sentence A such that $\mathcal{K} = Mod(\{A\})$ (since A can be the conjunction of sentences in Γ). (Note: *elementary* means first order.)

Definition: $\mathcal{K}$ is EC_Δ iff there is a set Γ (not necessarily finite) of sentences such that $\mathcal{K} = Mod(\Gamma)$.

Note that in either case,

$$\mathcal{K} = \bigcap_{A \in \Gamma} Mod(A).$$

Some examples:

The class of all groups in the language $(0, +, -)$ is an EC since Γ is the set of axioms for groups.

Consider the language with no non-logical symbols. The class of all infinite structures (those with infinite domains) is EC if there is a sentence (or finite set of sentences) that satisfies

$$\gamma_i = \exists x_1 \exists x_2 \ldots \exists x_i (\bigwedge_{1 \le j < k \le i} x_j \ne x_k)$$

for all i; i.e. there are $\ge i$ distinct objects. If we let $\Gamma = \{\gamma_2, \gamma_3, \ldots\}$, then $\mathcal{M} \models \Gamma$ iff M is infinite. So the class of infinite structures is EC_Δ. A fact which we will show later, is that the class of infinite structures is not an EC. Furthermore, there is no set Γ^* of sentences such that $\mathcal{M} \models \Gamma^*$ iff M is finite; so the class of finite structures is not even EC_Δ.

The class of infinite groups is EC_Δ since we can let

$$\Gamma = \{\gamma_2, \gamma_3, \ldots\} \cup \{\text{group axioms}\}.$$

(This class is not an EC.)

The class of fields with language $L = (0, 1, +, -)$ is an EC since there are a finite set of field axioms.

$\mathcal{K}_p$ = the class of fields of characteristic p, where p is a fixed prime, is an EC since we can let

$$\Gamma = \{\underbrace{1 + 1 + \ldots + 1}_{p} = 0\} \cup \{\text{field axioms}\}.$$

However, $\mathcal{K}_0$ = the class of fields of characteristic 0 is not an EC. It is an EC_Δ since we can let

$$\Gamma = \{\underbrace{1 + 1 + \ldots + 1}_{p} \neq 0 : p = 2, 3, 5, 7, \ldots\} \cup \{\text{field axioms}\}.$$

The class of fields of finite, non-zero characteristic is neither an EC nor EC_Δ. We'll show this later.

Q

Q is a theory about natural numbers due to Raphael Robinson. The language of Q is $(0, S, +, \cdot)$. The axioms for Q are

1. $\forall x(0 \neq Sx)$
2. $\forall x \forall y(x \neq y \rightarrow Sx \neq Sy)$
3. $\forall x(x \neq 0 \rightarrow \exists y(x = Sy)$
4. $\forall x(x + 0 = x)$
5. $\forall x \forall y(x + Sy = S(x + y))$
6. $\forall x(x \cdot 0 = 0)$
7. $\forall x \forall y(x \cdot Sy = x \cdot y + x)$

(Axioms 4-7 are inductive definitions for '+' and '$\cdot$'.) Q is the set of logical consequences of the above axioms.

Q has a model; $\mathcal{N} = (N, 0, S, +, \cdot)$ is such that $\mathcal{N} \models Q$. This is the standard model.

Q has another model, $\mathcal{A}$, with

$A = \{0, 1, 2, \ldots\} \cup \{\infty\}$
$0^{\mathcal{A}} = 0$
$S^{\mathcal{A}}(n) = n + 1$ for $n \in N$
$S^{\mathcal{A}}(\infty) = \infty$
$\infty +^{\mathcal{A}} \infty = \infty +^{\mathcal{A}} n = n +^{\mathcal{A}} \infty = \infty$ for $n \in N$
$\infty \cdot^{\mathcal{A}} 0 = 0 \cdot^{\mathcal{A}} \infty = 0$
$\infty \cdot^{\mathcal{A}} \infty = \infty \cdot^{\mathcal{A}} (n+1) = (n+1) \cdot^{\mathcal{A}} \infty = \infty$ for $n \in N$

So $\mathcal{A} \models Q$. A difference between $\mathcal{N}$ and $\mathcal{A}$ is that $\mathcal{A} \models \exists x(x = Sx)$, but $\mathcal{N} \models \forall x(x \neq Sx)$. So $Q \not\models \forall x(x \neq Sx)$.

Fact: $Q \not\models \forall x \forall y(x + y = y + x)$.
Proof: Homework. Try using the universe $N \cup \{\infty_1, \infty_2\}$.

Peano Arithmetic

Peano arithmetic (PA) is a set of sentences (a theory). The language of PA is $(0, S, \cdot)$. PA is the set of logical consequences of the axioms of Q plus the induction axioms. For every formula A, $A = A(x)$,

$$A(0/x) \wedge \forall x(A \to A(x+1/x)) \to \forall x A$$

is an axiom. A modified form of the induction axioms is for every formula A with free variables $x, y_1, \ldots, y_k$,

$$\forall y_1 \ldots \forall y_k (A(0/x) \wedge \forall x(A \to A(x+1/x)) \to \forall x A$$

is an axiom. Or in sloppier, but possibly clearer form,

$$\forall \vec{y}(A(0) \wedge \forall x(A(x) \to A(x+1)) \to \forall x A(x)).$$

For example, $PA \models \forall x(x \neq Sx)$.
Proof:

$$PA \models 0 \neq Sx$$

since $\forall x(0 \neq Sx)$ is an axiom.

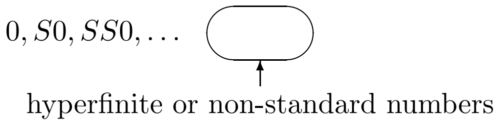
$$PA \models \forall x(x \neq Sx \to Sx \neq SSx)$$

since $\forall x \forall y (x \neq y \rightarrow Sx \neq Sy)$ is an axiom. Using the induction axiom for $x \neq Sx$, we have

$$PA \models 0 \neq S0 \wedge \forall x (x \neq Sx \rightarrow Sx \neq SSX) \rightarrow \forall x (x \neq Sx).$$

So $PA \models \forall x (x \neq Sx)$. □

PA has at least one model; the standard model $\mathcal{N} = (N, 0, S, +, \cdot)$. Fact: $PA \models \forall x \forall y (x + y = y + x)$. Any model (of Q also) has:



Validity of 1st Order Formulas is Undecidable

Math 260A - Mathematical Logic

November 30, 1988

(The treatment here is a little different from the text.) The major result is that given a formula, A, there is no effective procedure for determining whether or not $\models A$. The idea behind the proof is to express the halting problem as a first order formula in such a way that deciding the validity of the formula would solve the halting problem.

Let M be a Turing machine with states $q_0, \ldots, q_k$, and alphabet $a_0, \ldots$, a_ℓ, that only uses the right half of its input tape. Let q_0 be the starting state, and let q_1 be a special halting state; i.e. instead of halting, M goes to q_1 and loops forever. Let a_0 be the blank symbol. The halting problem for such a Turing machine is undecidable.

The first order language for such a Turing machine is $L = (0, S, Q_0, \ldots$, Q_k, $A_0, \ldots$, $A_\ell)$, where 0 is zero, S is the successor function, $Q_i(t, x)$ is a binary relation representing the fact that at time t, the the tape head is over the x^{th} tape square and M is in state q_i, and $A_i(t, x)$ is a binary relation representing the fact that at time t, the x^{th} tape square contains the symbol a_i.

We want a formula, C_M to say that M halts on the empty tape.

Preliminaries

a) To model the natural numbers, we need

$$\begin{aligned}
&\forall x(0 \neq Sx)\\
&\forall x \forall y(x \neq y \to Sx \neq Sy)\\
&\forall x(x \neq 0 \to \exists y(x = Sy))
\end{aligned}$$

b) To specify that the tape starts at the left most tape square and that the tape is initially blank, we need

$$Q_0(0,0)$$
$$\forall x A_0(0,x)$$

c) Now we need to specify that every tape square has at most one symbol:

$$\forall x \forall t(\neg(A_i(t,x) \wedge A_j(t,x))),\ i \neq j$$

that M is not in two states at once:

$$\forall x \forall t(\neg(Q_i(t,x) \wedge Q_j(t,x))),\ i \neq j$$

and that the tape head is not in two places at once:

$$\forall t \forall x \forall y(x \neq y \rightarrow \neg(\bigvee_{i=1}^{k} Q_i(t,x)) \wedge (\bigvee_{i=1}^{k} Q_i(t,y))).$$

d) We need to specify the instructions of M. For instructions of the form $q_i a_j a_m q_n$ which write symbols, we need:

$$\forall t \forall x(Q_i(t,x) \wedge A_j(t,x) \rightarrow Q_n(St,x) \wedge A_m(St,x)).$$

For instructions of the form $q_i a_j R q_n$ which move the tape head to the right, we need:

$$\forall t \forall x(Q_i(t,x) \wedge A_j(t,x) \rightarrow Q_n(St,Sx) \wedge A_j(St,x)).$$

And for instructions of the form $q_i a_j L q_n$ which move the tape head to the left, we need:

$$\forall t \forall x(Q_i(t,x) \wedge A_j(t,x) \rightarrow \exists y(Sy = x \wedge Q_n(St,y) \wedge A_j(St,x))).$$

(Remember that M never moves to the left of the initial square.)

e) Finally, we need to specify that the tape squares not under the tape head don't change:

$$\forall t \forall x[(\bigwedge_{i=1}^{k} \neg Q_i(t,x)) \rightarrow \bigwedge_{j=1}^{\ell} (A_j(t,x) \leftrightarrow A_j(St,x))].$$

Let the conjunction of the formulas in (a) - (e) be R_M. If $\mathcal{M} \models R_M$, then for standard times and positions (i.e. times and positions obtained by a finite number of applications of S and 0) the A_j and Q_i predicates correctly describe the operation of M on the empty tape. (Provided that M hasn't halted by time t.)

Proof: by induction on time.

To detect halting, we need the formula H_M:

$$\exists t \exists x (Q_1(t, x))$$

which says that M halts. (A problem that we'll come back to is "what if the t at which M halts is nonstandard?")

Now consider the formula C_M:

$$R_M \to H_M.$$

When is this valid? There are two cases to look at.

Case 1. M halts.

Claim: If M halts, then $\models R_M \to H_M$.

Proof: Take any structure, $\mathcal{A}$. If $\mathcal{A} \models R_M$, then for standard times and positions, $\mathcal{A} \models Q_i(t, x)$ iff M is in state q_i at time t at position x. So $\mathcal{A} \models Q_1(t_0, x_0)$ if M halts after t_0 steps at tape square x_0. ($Q_1(t_0, x_0)$ abbreviates $Q(S^{t_0}0, S^{x_0}0)$.)

Case 2. M doesn't halt. Consider two structures.

1. The standard model for R_M, $\mathcal{A}$:

   $A = \{0, 1, 2, \ldots\}$
   $S^{\mathcal{A}} = S$
   $Q_i^{\mathcal{A}}(n, m)$ - as determined by running M
   $A_j^{\mathcal{A}}(n, m)$ - as determined by running M

   Clearly, $\mathcal{A} \models R_M$, and $\mathcal{A} \not\models H_M$. So $\mathcal{A} \not\models R_M \to H_M$.

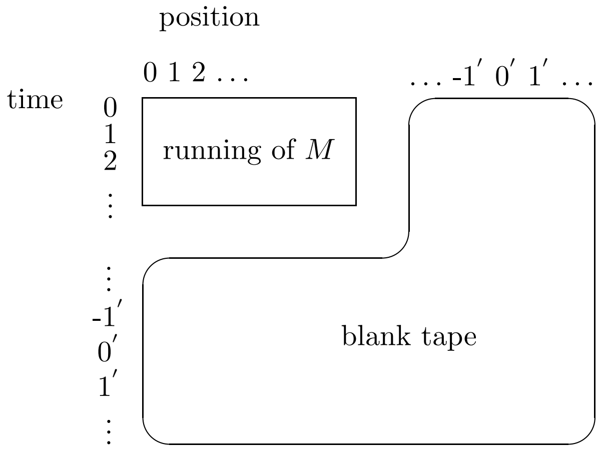
2. A nonstandard model, $\mathcal{B}$. Let

$$B = \{0, 1, 2, \ldots\} \cup \{\ldots, -2', -1', 0', 1', 2', \ldots\},$$

the disjoint union of the integers and the natural numbers. Also, let the interpretations of the non-logical symbols be (n, m are standard numbers, n', m' are non-standard numbers):

$$\begin{array}{l}
S^{\mathcal{B}}(n) = n + 1 \\
S^{\mathcal{B}}(n') = (n+1)' \\
Q_i^{\mathcal{B}}(n, m) \text{ - as before} \\
A_j^{\mathcal{B}}(n, m) \text{ - as before} \\
A_0^{\mathcal{B}}(x, m') \\
A_0^{\mathcal{B}}(n', x) \\
Q_1^{\mathcal{B}}(n', 0)
\end{array}$$

The last three sets of predicates state that the tape is blank for all non-standard times and positions, and that M is in a halting state at all non-standard times. If we view time and position pictorially, we have



Now, $\mathcal{B} \models R_M$ because at any time we only have one state and one position, so M transitions correctly. (At non-standard times, M is in a loop that has no beginning and no end.) So $\mathcal{B} \models R_M \wedge H_M$. So if M never halts, $R_M \rightarrow H_M$ is satisfiable, *but not valid*.

We have shown that $\models C_M$ iff M halts on the empty tape, so

Theorem: The set of valid first order formulas is not decidable.

More Undecidability, Databases

Math 260A - Mathematical Logic

December 2, 1988

Last time, we found, for a Turing machine M with extra restrictions (e.g. one way infinite tape, special halting state), a formula C_M such that $\models C_M$ iff M halts on the empty tape. An important point is that given M, we effectively get C_M.

Theorem: There is no effective procedure which, given a first order formula A, determines if $\models A$.

Proof: Suppose, for the sake of contradiction, that there was such an effective procedure. We could then determine if a given M halts on the empty tape by

1. forming C_M, and
2. using the effective procedure to see if C_M is valid. □

Now let's restrict our attention to finite structures. Is it decidable if a given first order formula A is true in every finite structure? No, but we have to change C_M. The reason we have to do this is that R_M codes M's computation using A_j, Q_i, S, and 0. So any model of R_M is infinite. We need to change R_M to allow finite structures. To do this, we'll change part (a) of the definition of R_M to

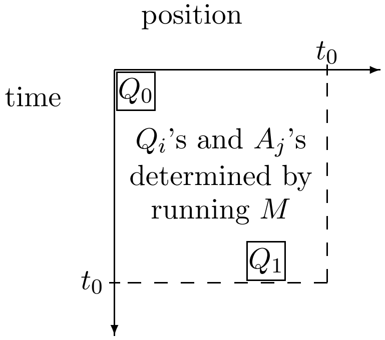
a$'$) We allow time to stop, and we force time to stop when M halts:

$$\begin{aligned}
&\forall x(0 \neq Sx)\\
&\forall x \exists y(x \neq 0 \rightarrow x = Sy)\\
&\forall x \forall y(Sx = Sy \rightarrow x = y \vee x = Sx \vee y = Sy)\\
&\forall x(x = Sx \leftrightarrow \exists y(Q_1(x, y)))
\end{aligned}$$

Now, let R^*_M be R_M with (a) changed to (a$'$).

Claim: R^*_M has a finite model iff M halts after starting on a blank tape.

Proof: If M halts in t_0 steps, it can't move right more than t_0 squares. So the model is finite; i.e.



If R^*_M has a finite model, then M has to halt since there must be a time t_0 such that $t_0 = St_0$. So R^*_M is satisfied by some finite structure iff M halts on the empty tape. □

Theorem: It is undecidable if a given first order formula has a finite model. (Note that R^*_M has a finite model iff it is not the case that $\neg R^*_M$ is true in every finite model.)

So it is undecidable if a given formula is true in every finite model.

Theorem: The set of formulas which have a finite model is r.e.

Proof: Given a first order formula A, the idea is to enumerate all finite structures in the language of A and check each one to see if it is a model of A. The following procedure does this:

```
for $i = 1, 2, \ldots$
    for each structure in the language with $i$ objects
        if the structure is a model of $A$ then
          halt
    end
end
```

(Note that for each i, there are only a finite number of structures with i objects in their universes (up to isomorphism) because each constant, function, and relation only has a finite domain and range. Also, note that we

can check to see if a structure is a model of A since the quantifiers only range over a finite universe.)
By Church's thesis, this procedure gives a partial recursive function whose domain is the set of formulas with finite models. Therefore this set is r.e. $\Box$

Corollary: The set of first order formulas which are true in every finite structure (i.e. the set of formulas whose negations do not have a finite model) is not r.e.

Proof: by contradiction. Given a formula, A, enumerate, in an interleaving fashion, the following sets:

1. all formulas satisfiable in a finite structure, and
2. all formulas true in every finite model.

Either A appears in list 2 and A is true in all finite models, or $\neg A$ appears in list 1 and A is not true in all finite models. This would then give us a decision procedure for deciding if a given formula is true in every finite model, a contradiction.

Databases

Definition: A *database* is a finite structure.

Definition: A language L is *relational* iff it has no function symbols. (We allow constant symbols.)

Definition: A *relational database* is a database in a relational language.

For example, "Fatherof(x, y)" is a relation and so could be in a relational database. But, "Father(y)" is a function and could not be in a relational database.

Older style databases have pointers between records which are essentially functions. Relational databases just have tables of relations.

Fact: (Codd) Basic query languages for relational databases can express exactly the first order formulas.

Some examples of queries. Let $L = \{E\}$ where E is a binary relation, and $E(x, y)$ iff there is an edge from x to y. A structure in the language L "is" a directed graph. To express the property "x is isolated", we would say

$$\forall y \neg E(x,y) \wedge \neg E(y,x)).$$

The relation $\{\langle x, y\rangle :$ there is a directed path from x to $y\}$ is not first order definable. (Because we have no way specifying that a path is constructed out of an arbitrary number of transitive operations.)

Proof Theory

Math 260B - Mathematical Logic

January 4, 1989

The overall business of mathematics is to determine whether or not certain first order formulas are valid. Why first order? Most of mathematics can be formalized in set theory, and set theory is usually formalized as a first order theory with nonlogical symbol $\in$ (i.e., $x \in y$). There are some gray areas; the axiom of choice and the continuum hypothesis are independent of set theory. The truth of these does not necessarily mean validity since there are structures in which they are false.

How do we show that something is valid? Proofs. Proofs are used to establish mathematical results. Our next task is to formalize the notion of proof. Proofs, as used by mathematicians, are a social phenomenon in the sense that they depend on what people are willing to accept. We want a mathematical definition of proof.

We will use a "refutation proof system". Instead of proving that ϕ is valid, we'll show that $\neg\phi$ is not satisfiable. More generally, given a set Γ of sentences, a refutation of Γ will show that Γ is not satisfiable. (An assumption that we'll make, which is not necessary for the development, is that first order languages are countable.)

The idea behind a refutation is to build a series of Γ_i's and end up with a Γ_n that is not satisfiable. Specifically, Γ_1 is Γ, and Γ_{i+1} will extend Γ_i by one more sentence in such a way that if Γ_i is satisfiable, then so is Γ_{i+1}. A refutation of Γ can be written out as the list of sentences added to make the Γ_i's. The property of being a refutation will be decidable.

Corollary: (Compactness Theorem) If Γ is unsatisfiable, then some finite subset is unsatisfiable.

Example: Let's show that

$$\exists x \forall y P(x,y) \models \forall y \exists x P(x,y) \qquad (*)$$

is valid. Let Γ be

$$\{\exists x \forall y P(x,y),\ \neg\forall y \exists x P(x,y)\}.$$

Then * is true iff Γ is not satisfiable. The first step is to put Γ into prenex normal form:

$$\Gamma' = \{\exists x \forall y P(x,y),\ \exists y \forall x \neg P(x,y)\}.$$

The next step is to refute Γ' with a derivation of a contradiction:

1.	$\exists x \forall y P(x,y)$	assumption
2.	$\exists y \forall x \neg P(x,y)$	assumption
3.	$\forall y P(a,y)$	existential instantiation from 1
4.	$\forall x \neg P(x,b)$	existential instantiation from 2
5.	$P(a,b)$	universal instantiation from 3
6.	$\neg P(a,b)$	universal instantiation from 4

Lines 5 and 6 are contradictory, hence * must be valid. (Existential and universal instantiation (EI and UI) give names for an *instance* of some variable. For example in line 3, we gave the name a to some x such that $\forall y P(x,y)$.)

Proof Theory (cont.)

Math 260B - Mathematical Logic

January 6, 1989

(continued from last lecture)

Example: In order to prove

$$\exists x \forall y (P(y) \leftrightarrow x = y) \models \exists x P(x),$$

we need a refutation of

$$\{\exists x \forall y (P(y) \leftrightarrow x = y), \neg \exists x P(x)\}.$$

1.	$\exists x \forall y (P(y) \leftrightarrow x = y)$	assumption
2.	$\forall x \neg P(x)$	assumption (in PNF)
3.	$\forall y (P(y) \leftrightarrow a = y)$	EI from 1
4.	$\neg P(a)$	UI from 2
5.	$P(a) \leftrightarrow a = a$	UI from 3
6.	$\forall x (x = x)$	equality axiom
7.	$a = a$	UI from 6

Since lines 4, 5, and 7 are tautologically inconsistent, the proof is complete. "Tautologically inconsistent" means that there is no way to assign truth values to the atomic subformulas $P(a)$ and $a = a$ that makes the formulas in lines 4, 5, and 7 all true.

Definition: Let Γ be a set of sentences with language L. Let L^+ be L plus new constant symbols $a_1, a_2, \ldots$. A *derivation* D from Γ is a sequence (finite or countably infinite) of sentences such that each sentence A in D satisfies one of the following:

1. A is a member of Γ. (A is an assumption.)

2. There is a sentence of the form $\exists x B$ in D occurring before A such that A is $B(a_i/x)$ for some a_i which has not yet occurred in D.

3. There is a sentence of the form $\forall x B$ in D occurring before A such that A is $B(t/x)$ for some term t in the language L^+. (We do allow t to have names a_j which haven't been used before.)

   Question: Why would we want to substitute terms instead of just names?

   Answer: Suppose we wanted to show that $\models \exists x(S0 = x)$. The following refutation does this.

1.	$\forall x(\neg S0 = x)$	assumption (in PNF)
2.	$\neg S0 = S0$	UI
3.	$\forall x(x = x)$	equality axiom
4.	$S0 = S0$	UI

   Note that step 2 substitutes the term $S0$ for the quantified variable.

4. A is one of the following equality axioms:

   (a) (reflexivity) $\forall x(x = x)$.

   (b) (symmetry) $\forall x \forall y(x = y \rightarrow y = x)$.

   (c) (transitivity) $\forall x \forall y \forall z(x = y \wedge y = z \rightarrow x = z)$.

   (d) For each k-ary predicate symbol P in L,

   $$\forall x_1 \ldots \forall x_k \forall y_1 \ldots \forall y_k (\bigwedge_{i=1}^{k} x_i = y_i \wedge P(x_1, \ldots, x_k) \rightarrow P(y_1, \ldots, y_k))$$

   is an equality axiom.

   (e) For each k-ary function symbol f in L,

   $$\forall x_1 \ldots \forall x_k \forall y_1 \ldots \forall y_k (\bigwedge_{i=1}^{k} x_i = y_i \rightarrow f(x_1, \ldots, x_k) = f(y_1, \ldots, y_k))$$

   is an equality axiom.

> *Question:* Why do we need d) and e)?
>
> *Answer:* Consider the following structure with P a unary relation, and $=$ a binary relation:
>
> > universe $= \{0, 1\}$,
> > $0 = 0$, $0 = 1$, $1 = 0$, $1 = 1$, and
> > $P(0)$, $\neg P(1)$.
>
> d) and e) constrain structures to use $=$ as true equals. When we were discussing model theory, we defined $=$ as true equals and didn't leave it up for interpretation; d) and e) provide the same constraints in proof theory.

Definition: A *refutation* of Γ is a finite length derivation D from Γ such that the set of quantifier free sentences in D is tautologically inconsistent.

Proposition: If Γ is decidable (i.e. we can determine whether or not a sentence is in Γ), then the property of being a refutation of Γ is decidable. (I.e. there is an effective algorithm which determines whether or not a set of sentences is a refutation of Γ.

> *Question:* What if there are infinitely many function or predicate symbols?
>
> *Answer*: One convention is to assume that the number of function or predicate symbols of arity k is computable. (Since we usually know what language we're using, we'll assume this convention.) Another convention is to require that terms in UI instances and equality axioms only contain function/predicate/constant symbols that have already appeared in an assumption of Γ. (Note that this alternate convention will require a change in the definition of derivation.)

Observation: We can Gödel number sentences, so we can Gödel number derivations, and so we can Gödel number refutations. So a refutation is just a string of symbols that follow certain rules.

Theorem: (soundness) If Γ has a refutation, then Γ is not satisfiable.

Theorem: (completeness, Gödel, 1929) If Γ is not satisfiable, then Γ has a refutation.

These two theorems give us the fact that Γ has a model iff there is no refutation of Γ. Or, in other words, ϕ is valid ($\models \phi$) iff there is a refutation of $\neg\phi$. This provides us with a link between being true in all structures and being a refutation of $\neg\phi$. Or, in other words, a link between syntax and semantics.

Notation: $\vdash \phi$ means that ϕ is provable (or that $\neg\phi$ has a refutation). $\Gamma \vdash \phi$ means that ϕ is provable from Γ (or that $\Gamma \cup \{\neg\phi\}$ has a refutation).

Gödel says that $\vdash \phi \ \Leftrightarrow \models \phi$.

Soundness and Completeness

Math 260B - Mathematical Logic

January 9, 1989

A few notes about derivations: Sometimes derivations will be labeled with justifications; i.e. "EI (n)" or "UI (n)". These comments denote that a given line was inferred from line number n by EI or UI. Assumptions must be in prenex normal form, and may have to be labeled with the original formula in order to make derivations decidable.

Theorem: (Soundness.) If Γ has a refutation, then Γ is unsatisfiable.

Proof: Let the language of Γ be L, and let D be a refutation of Γ in the language L^+.

> **Lemma**: If $\mathcal{A}_i$ is a structure which makes Γ and the first i sentences in D true, then there exists an $\mathcal{A}_{i+1}$ that makes Γ and the first $i+1$ sentences of D true.
>
> *Proof*: (Without loss of generality, assume that the language of $\mathcal{A}_j$ contains exactly the symbols from Γ and the first j lines of D.) Let ϕ be the $(i+1)^{st}$ sentence in D.
>
> **case 1.** $\phi \in \Gamma$. Then set $\mathcal{A}_{i+1} = \mathcal{A}_i$.
>
> **case 2.** ϕ is an equality axiom. Then set $\mathcal{A}_{i+1} = \mathcal{A}_i$.
>
> **case 3.** ϕ is obtained by EI from $\exists x \psi$ appearing earlier in D. Then ϕ is $\psi(a_n/x)$ for some new a_n. Since $\mathcal{A}_i \models \exists x \psi$, there is a $b \in |\mathcal{A}_i|$ such that $\mathcal{A}_i \models \psi(b/x)$. Form $\mathcal{A}_{i+1}$ by taking $\mathcal{A}_i$ and letting $a_n^{\mathcal{A}_{i+1}} = b$.
>
> **case 4.** ϕ is $\psi(t/x)$ derived by UI from $\forall x \psi$ which appeared earlier in D. If t involves no new constant symbols, take $\mathcal{A}_{i+1} = \mathcal{A}_i$. (Remember, derivations can't introduce new

> function or predicate symbols.) If t has new constant symbols, say $a_n, a_{n+1}, \ldots$, then let b be an arbitrary element of $|\mathcal{A}_i|$, and let $\mathcal{A}_{i+1}$ be $\mathcal{A}_i$ plus $a_n^{\mathcal{A}_{i+1}} = b$, $a_{n+1}^{\mathcal{A}_{i+1}} = b, \ldots$

Note that in constructing $\mathcal{A}_{i+1}$, we never change $|\mathcal{A}_i|$; we only add new constant symbols. □

The soundness theorem is immediate from the lemma. (The lemma proves the contrapositive of the theorem.) □

Theorem: (Completeness.) If Γ is unsatisfiable, then Γ has a refutation; or (Gödel) if there is no refutation of Γ, then Γ has a model. (Note that the completeness theorem is true for uncountable Γ, but we'll only prove it for countable Γ.

Before proving the completeness theorem, we'll develop the notion of canonical derivation.

Definition: Assuming that Γ is countable, a derivation D is *canonical* iff

1. Every sentence in Γ appears in D.
2. Every equality axiom in the language L of Γ appears in D.
3. For every sentence of the form $\exists x \psi$ in D, there is another sentence $\psi(t/x)$ for some term[1] t in L^+.
4. For every sentence of the form $\forall x \psi$ in D and every term t involving only symbols from L and constants appearing in D, $\psi(t/x)$ is in D. (Note that we mean *all* terms t, not just those appearing earlier in D.)
5. If $\forall x \psi$ is in D, then $\psi(t/x)$ is in D for at least one t. (This is just to make sure that there is at least one constant in the language.)

> *Question:* Why do we need at least one constant in the L?
> *Answer:* If we have a universally quantified formula, we need to make sure that there is something to quantify over. This is basically the same reason that we require structures to have non-empty domains.

[1] The terms that appear in canonical derivations are "closed" terms; terms with no variables.

Example: Let $\Gamma = \{\exists x(f(x) = f(x) \wedge f(x) \neq f(x))\}$. The terms of Γ are:

$$a_1,\ f(a_1),\ f(f(a_1)), \ldots$$

The atomic formulas of Γ are

$$\begin{array}{rcc} a_1 = a_1, & a_1 = f(a_1), & a_1 = f(f(a_1)), \ldots \\ f(a_1) = a_1, & f(a_1) = f(a_1), & f(a_1) = f(f(a_1)), \ldots \\ f(f(a_1)) = a_1, & \ldots & \\ \ldots & & \end{array}$$

A canonical derivation contains instances of the equality axioms:

$a_1 = a_1,\ f(a_1) = f(a_1), \ldots$
$a_1 = a_1 \rightarrow a_1 = a_1,\ a_1 = f(a_1) \rightarrow f(a_1) = a_1, \ldots$
$a_1 = a_1 \wedge a_1 = a_1 \rightarrow a_1 = a_1,\ a_1 = f(a_1) \wedge f(a_1) = f(f(a_1)) \rightarrow a_1 = f(f(a_1)), \ldots$
$a_1 = a_1 \rightarrow f(a_1) = f(a_1),\ a_1 = f(a_1) \rightarrow f(a_1) = f(f(a_1)), \ldots$

Proposition: Let Γ be a countable set of sentences. Then there is a canonical derivation from Γ.
Proof: Let $\Gamma = \{\gamma_1, \gamma_2, \ldots\}$. Let D_0 be the three equality axioms:

$$\begin{array}{l} \forall x(x = x) \\ \forall x \forall y(x = y \rightarrow y = x) \\ \forall x \forall y \forall z(x = y \wedge y = z \rightarrow x = z) \end{array}$$

For $i > 0$, let D_i be D_{i-1} plus:

1. γ_i if there is one.

2. The equality axioms for any function and/or predicate symbols that just appeared for the first time in γ_i.

3. For any sentence of the form $\exists x\psi$ in D_{i-1} that hasn't already been EI'ed, add $\psi(a_n/x)$ for some new a_n.

4. For every sentence of the form $\forall x\psi$ in D_{i-1} and every term t with less than or equal to i occurrences of function symbols in the language L plus the constants used so far, add $\psi(t/x)$. If no such t exists, add $\psi(a_1/x)$.

5. For the sake of efficiency, don't add sentences that have already been added.

Let D be the limit of $D_0, D_1, \ldots$ (i.e. $D = \bigcup D_i$.) □

We'll use the following to show completeness:

Main Lemma: Suppose that D is a canonical derivation from Γ and that no finite initial part of D is a refutation of Γ (i.e., no finite set of quantifier free sentences in D is tautologically inconsistent). Then Γ is satisfiable.

Proof: Next time.

> *Question:* How do we know that there isn't a non-canonical derivation which refutes Γ?
>
> *Answer:* The lemma shows that if D doesn't refute Γ, then Γ is satisfiable. The soundness theorem then says that since Γ is satisfiable, Γ has no refutation. So the lemma is actually stronger than the assertion that if there is no refutation of Γ, then Γ is satisfiable.

Note that given an effectively enumerable set Γ, we can search for a refutation by following the algorithm in the proof of the above proposition (and checking for tautological inconsistency after each D_i). This will find a refutation iff one exists. This is an improvement over the brute force search which codes derivations and first looks at all strings of length 1, then of length 2, and so on.

Corollary: (to Gödel completeness theorem.) The set of valid formulas is r.e.

Proof: A formula ϕ is valid iff $\neg\phi$ has a refutation. We can list valid formulas by generating canonical derivations. Whenever we find a refutation of $\neg\phi$ for any ϕ, write out ϕ. □

Completeness

Math 260B - Mathematical Logic

January 11, 1989

Main Lemma: If there is a canonical derivation D from Γ (countable) such that every finite set of quantifier free sentences in D is tautologically consistent, then Γ is satisfiable.

Proof: Let L_2 be the language L plus the a_n's used in D. (L_2 may be all of L^+ if every a_n in L^+ is used in D.) Consider the set of atomic formulas in the language of L_2 as propositional variables. By the compactness theorem for propositional logic, there is an assignment σ of truth values to the atomic L_2 formulas such that σ makes every quantifier free sentence in D true. In order to finish the proof of the main lemma and the completeness theorem, we'll need the following four lemmas.

Definition: If r and s are L_2 terms, $r \sim s$ means that $\sigma(r = s)$ = true; i.e., σ assigns the value true to the atomic formula $r = s$.

Lemma 1: $\sim$ is an equivalence relation.

Proof: We must show that $\sim$ is reflexive, symmetric, and transitive.

Reflexivity. Since $\forall x(x = x)$ is an equality axiom, it must appear in any canonical derivation, D. Hence $r = r$ is in D for all terms r. So $\sigma(r = r)$ = true for all terms r.

Symmetry. Since $\forall x \forall y(x = y \to y = x)$ is an equality axiom, it must appear in any canonical derivation, D. Hence $r = s \to s = r$ is in D for all terms r and s. So σ makes all of these pairs of atomic formulas true. Hence $\sim$ is symmetric.

Transitivity. Since $\forall x \forall y \forall z(x = y \wedge y = z \to x = z)$ is an equality axiom, it must appear in any canonical derivation, D. Hence $r = s \wedge s = t \to r = t$ is in D for all terms r, s and t. So σ makes all of these triples of atomic formulas true. Hence $\sim$ is transitive. □

Lemma 2: ($\sim$ respects the function and predicate symbols of L.) If f is a k-ary function symbol in L and if $r_1 \sim s_1, \ldots, r_k \sim s_k$, then $f(r_1, \ldots, r_k) \sim f(s_1, \ldots, s_k)$. If P is a k-ary predicate symbol in L and if $r_1 \sim s_1, \ldots, r_k \sim s_k$, then $\sigma(P(r_1, \ldots, r_k)) = \sigma(P(s_1, \ldots, s_k))$.

Proof: By the equality axioms,

$$\sigma(r_1 = s_1 \wedge \ldots \wedge r_k = s_k \to f(r_1, \ldots, r_k) = f(s_1, \ldots, s_k)) = \text{true},$$
$$\sigma(r_1 = s_1 \wedge \ldots \wedge r_k = s_k \wedge P(r_1, \ldots, r_k) \to P(s_1, \ldots, s_k)) = \text{true}$$

for all $\vec{r}$ and $\vec{s}$. □

Definition: $[r] = \{s : r \sim s\}$.

Definition: Let $\mathcal{A}$ be a structure in the language L_2 (the language of D) defined as follows:

1. $|\mathcal{A}| = \{[r] : r \text{ is an } L_2 \text{ term}\}$.[1]
2. For every function symbol f in L_2, $f^{\mathcal{A}}$ is defined by
$$f^{\mathcal{A}}([r_1], \ldots, [r_k]) = [f(r_1, \ldots, r_k)].$$
3. For every predicate symbol P in L_2, $P^{\mathcal{A}}$ is defined by
$$P^{\mathcal{A}}([r_1], \ldots, [r_k]) \text{ iff } \sigma(P(r_1, \ldots, r_k)) = \text{true}.$$
4. For every constant symbol c in L_2, $c^{\mathcal{A}} = [c]$.

[1] A slight modification to $|\mathcal{A}|$ is made at the beginning of the next lecture.

Lemma 3: $\mathcal{A}$ is well defined as a structure.

Proof: By Lemma 2. □

Lemma 4: $\mathcal{A} \models \phi$ for every ϕ in D.

Proof:

> **Claim 1:** For any term r, let $r^{\mathcal{A}}$ be r with each constant symbol c changed to $c^{\mathcal{A}}$ and each function symbol f changed to $f^{\mathcal{A}}$. Then $r^{\mathcal{A}} = [r]$.
>
> *Proof*: By induction on the complexity of r and parts 2 and 4 of the definition of $\mathcal{A}$. □
>
> **Claim 2:** Let ϕ be an atomic formula. Then $\mathcal{A} \models \phi$ iff $\sigma(\phi) =$ true.
>
> *Proof*: Obvious by part 3 of the definition of $\mathcal{A}$ and a slight variant of claim 1. □

(Proof of Lemma 4 continued.) By induction on the number of quantifiers in ϕ.

Case 1. ϕ is quantifier free. Since $\sigma(\phi) =$ true and by claim 2, σ and $\mathcal{A}$ agree on the truth of atomic subformulas of ϕ.

Case 2. If ϕ is of the form $\exists x\psi$, then for some term t, $\psi(t/x)$ is in D. By the induction hypothesis, $\mathcal{A} \models \psi(t/x)$. Hence, by claim 1, $\mathcal{A} \models \exists x\psi$.

Case 3. ϕ is of the form $\forall x\psi$. For every L_2 term r, $\psi(r/x)$ is in D. By the induction hypothesis, $\mathcal{A} \models \psi(r/x)$ for all terms r. Hence, by claim 1 and induction on the complexity of r, $\mathcal{A} \models \psi([r]/x)$ for all terms r. $\mathcal{A} \models \psi(r^{\mathcal{A}}/x)$ follows from $\mathcal{A} \models \psi(r/x)$. Hence for all $a \in |\mathcal{A}|$, $\mathcal{A} \models \psi(a/x)$. So $\mathcal{A} \models \forall x\psi$ by the definition of truth.[2] □

(Proof of Main Lemma and Gödel's completeness theorem continued.) Since D is canonical, $D \supseteq \Gamma$. So $\mathcal{A} \models \phi$ for all $\phi \in \Gamma$. By letting $\mathcal{A}^-$ be $\mathcal{A}$ restricted to the language of Γ (i.e. eliminate new constant symbols), $\mathcal{A}^- \models \Gamma$. □

[2] For more details, see the beginning of the next lecture.

Question: Why do we need to restrict $\mathcal{A}$ to $\mathcal{A}^-$?

Answer: We don't really. But last quarter we gave two definitions of what it meant for a structure $\mathcal{A}$ to satisfy a set of sentences Γ: one admitted the possibility of extra constant symbols, and the other didn't.

Corollaries of Completeness

Math 260B - Mathematical Logic

January 13, 1989

Definition: r is a *closed* term if it contains no variables.

Definition: If r is closed, then $r^{\mathcal{A}}$ denotes the value of the term r in $\mathcal{A}$.

Recall, from the definition of truth, that if s is an object assignment, then $\bar{s}(r)$ denotes an object. If r is closed, then $\bar{s}(r)$ is independent of s. So $r^{\mathcal{A}} \overset{\text{def}}{=} \bar{s}(r)$ for any object assignment s.

Last time, we defined

$$|\mathcal{A}| = \{[r] : r \text{ is a closed term in } L_2\}.$$

r was comprised of constant symbols in D and function symbols in L, the language of Γ.

Definition: $\mathcal{A}$ is a *Herbrand model* because every object in $|\mathcal{A}|$ is equal to $r^{\mathcal{A}}$ for some closed term r in L_2, the language of $\mathcal{A}$.

In case 3 of lemma 4, we showed that if $\phi = \forall x\psi$ is in D, then $\mathcal{A} \models \phi$. The reason was that for all $a \in |\mathcal{A}|$, $a = r^{\mathcal{A}} = [r]$ for some closed term r. Since $\psi(r/x)$ is in D, $\mathcal{A} \models \psi(r^{\mathcal{A}}/x)$. (i.e. $\mathcal{A} \models \psi[s]$ for $s(x) \rightarrow r^{\mathcal{A}}$.) By induction, $\mathcal{A} \models \psi(r^{\mathcal{A}}/x)$ for all $r^{\mathcal{A}}$. So $\mathcal{A} \models \forall x\psi$ by the definition of truth.

We also defined $\mathcal{A}^-$ with the language L (not L_2) to be

$$\begin{aligned} |\mathcal{A}^-| &= |\mathcal{A}|, \\ P^{\mathcal{A}^-} &= P^{\mathcal{A}}, \\ f^{\mathcal{A}^-} &= f^{\mathcal{A}}, \text{ and} \\ c^{\mathcal{A}^-} &= c^{\mathcal{A}} \end{aligned}$$

for all $P, f, c \in L$. And we showed that $\mathcal{A}^- \models \Gamma$ since $\mathcal{A} \models \Gamma$. Even though we threw away constant symbols from L_2, we didn't throw away the objects that those symbols represented. Note that because we threw away some constant symbols, $\mathcal{A}^-$ may not be a Herbrand model in the language of L.

> *Question:* Could we throw away those objects and still have a model of Γ?
>
> *Answer:* Not in general. For example, suppose that $\Gamma = \{\exists x P(x),\ P(c)\}$. If $\mathcal{A}$ had only one object in $P^{\mathcal{A}}$ and we threw it away, then $\mathcal{A} \not\models \exists x P(x)$.

The following 5 corollaries result from Gödel's completeness theorem.

Corollary 1: If Γ is a countable set of sentences and Γ is satisfiable, then Γ has a countable (possibly finite) model.

Proof: The $\mathcal{A}^-$ in the proof of the completeness theorem is countable because we started with a countable language and added countably many constant symbols. □

Corollary 2: (compactness theorem) If Γ is unsatisfiable, then some finite subset of Γ is unsatisfiable

Proof: Since Γ is unsatisfiable, it has a refutation, D. Since refutations are finite, D uses a finite number of sentences from Γ. This is then an unsatisfiable finite subset of Γ. □

Corollary 3: If $\Gamma \vdash \phi$, then for some finite $\Gamma_0 \subseteq \Gamma$, $\Gamma_0 \vdash \phi$.

Proof: Since $\Gamma \vdash \phi$, $\Gamma \cup \{\neg\phi\}$ is unsatisfiable. By corollary 2, there is some finite subset Γ_0 such that $\Gamma_0 \cup \{\neg\phi\}$ is unsatisfiable. Hence $\Gamma_0 \vdash \phi$. □

Corollary 4: If Γ has arbitrarily large finite models, then Γ has an infinite model.

Proof: Let $\Delta = \Gamma \cup \{\alpha_1, \alpha_2, \ldots\}$, where

$$\alpha_k = \exists x_1 \ldots \exists x_k (\bigwedge_{1 \le i < j \le k} x_i \neq x_j);$$

i.e. α_k says that there are at least k distinct objects. Now every finite subset of Δ has a model, so by the compactness theorem, Δ has a model which is necessarily infinite. □

As an example, consider the following definition.

Definition: The *theory of* $\mathcal{R}$, $Th(\mathcal{R})$, where $\mathcal{R} = (R, 0, +, \cdot)$, is the set of first order sentences true in $\mathcal{R}$.

By corollary 1, $Th(\mathcal{R})$ has a countable model since $Th(\mathcal{R})$ is a countable set of sentences. Later, we'll see that *ZF*, the usual set theory, is a first order theory and is presumed to be satisfiable. Hence *ZF* has a countable model. Such a model will then satisfy the sentence which asserts that "R is uncountable"; i.e. "$\neg\exists$ a 1-1 map from N onto R". This sentence will be true in the structure, but false in the real universe. (This is Skolem's paradox.)

We have shown:

- that the set of formulas valid in all structures is r.e. but not recursive and hence not co-r.e. and
- that the set of formulas valid in all finite structures is co-r.e. but not recursive and hence not r.e. (We did this last quarter.)

Corollary 5: There is no Γ in any language L such that the models of Γ are precisely the finite structures in the language L.

Proof: Similar to the proof of corollary 4. Suppose that $\mathcal{A} \models \Gamma$ iff $|\mathcal{A}|$ is finite. Let α_j say that there are at least j objects. Now $\Gamma \cup \{\alpha_1, \ldots, \alpha_i\}$ has arbitrarily large models, and hence by corollary 4 has an infinite model. Such a model is also a model of Γ itself. □

We can now prove the existence of nonstandard models of arithmetic. Let $Th(N, 0, S, +, \cdot)$ be the set of sentences true in $(N, 0, S, +, \cdot)$. Then $Th(N, 0, S, +, \cdot)$ has a nonstandard countable model.

Proof: Let Γ be

$$Th(N, 0, S, +, \cdot) \cup \{c \neq 0,\ c \neq 1, \ldots\}$$

where c is a new constant symbol. Every finite subset of Γ is satisfiable. So Γ has a countable model $\mathcal{A}$ from which we can discard the constant symbol c (but not the object it denotes) to get the nonstandard model of $Th(N, 0, S, +, \cdot)$. (Note that $\mathcal{A} \models c > S^k 0$ for all $k \in N$, so c is an "infinite" element.) □

Set Theory

Math 260B - Mathematical Logic

January 18, 1989

Today's lecture will introduce Zermelo-Fraenkel set theory (ZF). This is a language whose only non-logical symbol is '$\in$'. The objects of ZF are intended to represent sets. The following formulas are axioms of ZF.

Null Set Axiom: There is a set with nothing in it:

$$\exists x \forall y (y \notin x).$$

Pair Set Axiom: For any two objects x and y, there is a set $z = \{x, y\}$ that contains exactly those two objects:

$$\forall x \forall y \exists z \forall u (u \in z \leftrightarrow u = x \lor u = y).$$

Union Axiom: For any set x, there is another set y, such that y is the union of all objects in x; i.e. $y = \bigcup_{z \in x} z$:

$$\forall x \exists y \forall z (z \in y \leftrightarrow \exists w (w \in x \land z \in w)).$$

Extensionality Axiom: If, for any two sets x and y, all sets in x are also in y and vice versa, then x is the same as y:

$$\forall x \forall y (\forall z (z \in x \leftrightarrow z \in y) \rightarrow x = y).$$

This prevents the existence of *urelements*, which are sets with nothing in them and which are not the empty set.

Regularity Axiom: Every set (except the empty set) has a "least" element; least in the sense that we can't have $\ldots x_3 \in x_2 \in x_1$:

$$\forall x(x \neq \emptyset \rightarrow \exists y(y \in x \wedge \forall z(z \in x \rightarrow z \notin y)).$$

Note that this disallows having $\{x\} \in \{x\}$. Also note the occurrence of $\emptyset$. We can define $\emptyset$, $x \cap y$, $x \cup y$, and $x \subseteq y$, in terms of first order set theory.

Subset Axiom: For any set x and formula A, there is another set y consisting of all members of x that make A true; i.e. $y = \{z \in x : A(z)\}$:

$$\forall x \exists y(\forall z(z \in y \leftrightarrow A(z) \wedge z \in x)).$$

Note that this is an axiom schema since A can be any formula.

Power Set Axiom: For any set x, there is a power set y of x; i.e., y contains all subsets of x:

$$\forall x \exists y \forall z(z \in y \leftrightarrow z \subseteq x).$$

Note that this axiom doesn't follow from the subset axiom. Also note that "$z \subseteq x$" is an abbreviation for $\forall w(w \in z \rightarrow w \in x)$.

Infinity Axiom: There is a set containing an infinite number of elements:

$$\exists x(\emptyset \in x \wedge \forall y(y \in x \rightarrow \{y\} \in x)).$$

Note that the existence of $\{y\}$ is implied by the pair set axiom; i.e. $\{y, y\}$.

Replacement Axiom: If (the graph of) a function can be defined by a first order formula A, and A's domain is a set, then its range is a set which can be constructed by replacing each element in the domain by the corresponding element in the range of A.

$$\forall w[\forall x(x \in w \rightarrow \exists! y A(x,y)) \rightarrow \exists v \forall x(x \in w \rightarrow \exists y(y \in v \wedge A(x,y)))].$$

Note the occurrence of "$\exists!$". This means that there exists a unique object. $\exists! y A(x,y)$ can be defined as

$$\exists y A(x,y) \wedge \forall y \forall y^{'}(A(x,y) \wedge A(x,y^{'}) \rightarrow y = y^{'}).$$

Axiom of Choice: ZFC is the above set of axioms together with the axiom of choice. The axiom of choice says that if we have a set of nonempty sets, we can choose a set of representatives of those sets:

$$\begin{aligned}\forall x[\emptyset \notin x \quad \rightarrow \quad & \exists v(v \text{ is a set of ordered pairs} \\ & \text{coding a 1-1 function } f \text{ with domain } x \\ & \text{and such that } \forall w \in x \rightarrow f(w) \in w)].\end{aligned}$$

Explanation: x is a set of sets. w is a set in x. f is the choice function. f takes as input an element w from x and produces an element from w.

An "ordered pair" $\langle u, v\rangle$ is defined by $\{\{u\}, \{u, v\}\}$ using three applications of the pair set axiom. Ordered pairs are such that

$$\forall u \forall v \forall u^{'} \forall v^{'} \langle u, v\rangle = \langle u^{'}, v^{'}\rangle \rightarrow u = u^{'} \wedge v = v^{'}.$$

"x is an ordered pair" is defined by $\exists u \exists v(x = \langle u, v\rangle)$. The domain of a set of ordered pairs z is defined by

$$\{u : \exists v \langle u, v\rangle \in z \wedge u \in \bigcup_{r \in \bigcup_{s \in z} s} r\}.$$

Explanation: z is a set of ordered pairs $\{\langle u_1, v_1\rangle, \langle u_2, v_2\rangle, \ldots, \langle u_k, v_k\rangle\}$. $\bigcup_{s \in z} s$ is the union of the ordered pairs. The r's are ordered pairs. $\bigcup_{r \in \bigcup_{s \in z} s} r$ is the union of the elements of those ordered pairs; i.e., $\bigcup_{r \in \bigcup_{s \in z} s} r = \{u_1, v_1, u_2, v_2, \ldots, u_k, v_k\}$. u is one of those elements. The double union essentially "strips away the braces" to get at the elements of the ordered pairs.

The range is similarly defined, and from those, function can be defined.

Ordinals and Cardinals

Math 260B - Mathematical Logic

January 20, 1989

Ordinals can be thought of as "canonical well-ordered sets". The integers can be represented by sets as follows:

$$\begin{array}{rll} 0 & \text{is denoted by} & \emptyset \\ 1 & \text{is denoted by} & \{\emptyset\} \\ 2 & \text{is denoted by} & \{\emptyset, \{\emptyset\}\} \\ 3 & \text{is denoted by} & \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\} \\ & \vdots & \\ n+1 & \text{is denoted by} & n \cup \{n\} \\ & \vdots & \end{array}$$

With this convention, $i < j$ iff $i \in j$ for any integers i and j. The set $\{0, 1, 2, \ldots\} \overset{\text{def}}{=} \omega \overset{\text{def}}{=} N$. Using the convention for representing integers, $\omega + 1 = \omega \cup \{\omega\}$.

Definition: A binary relation $\prec$ *well-orders* a set A iff $\prec$ is a linear total ordering of A and for every nonempty subset B of A there is a minimal element of B in the sense that $\exists x \in B (\forall y \in B (x \neq y \rightarrow x \prec y))$.

Definition: A set A is *transitive* iff $\forall x \forall y (y \in x \land x \in A \rightarrow y \in A)$.

Definition: A set is an *ordinal* iff it is transitive and well-ordered by $\in$.

Example: $\{0, 2\}$ is well-ordered by $\in$, but it is not transitive since $1 \in 2$. So it is not an ordinal.

Theorem: Every well-ordered set A (with order $\prec$) is isomorphic to some ordinal α (with order $\in$); i.e. $(A, \prec)$ is isomorphic to $(\alpha, \in)$.

Proof: (outline) Let the elements of A be ordered as $\{a_0 \prec a_1 \prec \ldots a_k \prec a^{'} \prec \ldots\}$. $a^{'}$ is *nice* iff $(\{a \in A : a \prec a^{'}\}, \prec)$ is isomorphic to some ordinal α with order $\in$; i.e. iff there is an isomorphism of $a_0, \ldots a_k$ to some ordinal α with order $\in$. If there is a non-nice symbol, then there is a minimal one. From that, we can derive a contradiction. □

The following are some examples of the properties of ordinals.

Example: $\omega + \omega :\ \underbrace{0 \prec 1 \prec 2 \ldots}_{\omega}\ \underbrace{0^{'} \prec 1^{'} \prec 2^{'} \ldots}_{\omega}$

Example: $1 + \omega = \omega :\ 1 \prec \underbrace{0^{'} \prec 1^{'} \prec 2^{'} \ldots}_{\omega}$

This example, combined with the earler definition of $\omega + 1$, leads us to think of $+$ as "followed by" when applied to ordinals .

Example: $(\{0, 2\}, \in)$ is isomorphic to $2 = (\{0, 1\}, \in)$.

Definition: (of $+$ applied to ordinals) Let α_1 and α_2 be ordinals. Let $(\alpha_2^{'}, \prec)$ be an isomorphic copy of $(\alpha_2, \in)$ such that $\alpha_2^{'} \cap \alpha_1 = \emptyset$. Let $\mathcal{A} = (\alpha_1 \cup \alpha_2^{'}, <)$ where $<$ is $\in$ on α_1, $<$ is $\prec$ on $\alpha_2^{'}$, and $a < a^{'}$ for all $a \in \alpha_1$ and $a^{'} \in \alpha_2^{'}$. Then $\alpha_1 + \alpha_2$ is the ordinal that $\mathcal{A}$ is isomorphic to.

Fact: For all ordinals α and β, either $\alpha = \beta$, $\alpha \in \beta$, or $\beta \in \alpha$.

Theorem: (depends on axiom of choice) Every set can be well-ordered.

The cardinality of a set is the size of the set. A cardinal will be a canonical set of a given size. $Card(x)$ will denote the cardinality of x.

Definition: $Card(x) \leq Card(y)$ iff there is a 1-1 function $f : x \rightarrow y$.

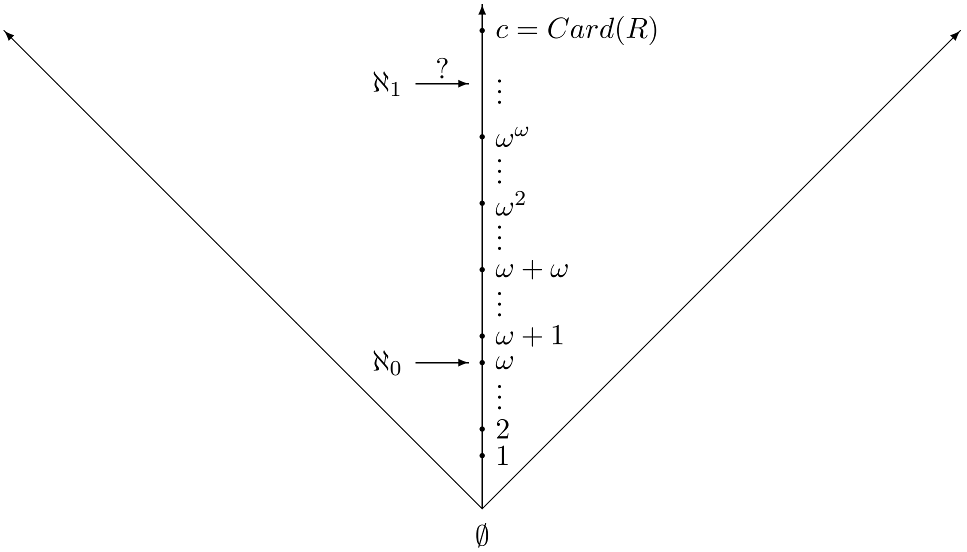
Theorem: [Schroeder-Bernstein] (depends on axiom of choice) If $Card(x) \leq Card(y)$ and $Card(y) \leq Card(x)$, then there is a 1-1 *onto* function $f : x \rightarrow y$; i.e. x and y are isomorphic sets.

Theorem: (depends on axiom of choice) For any 2 sets x and y, either $Card(x) \leq Card(y)$ or $Card(y) \leq Card(x)$.

Proof: Since x and y are well-orderable. □

Definition: x is a *cardinal* iff x is an ordinal and $Card(x) \neq Card(y)$ for all $y < x$.

A cardinal is the least (in the sense of $\in$) ordinal of a given cardinality. Pictorially, we have the following universe of all sets:



Universe of All Sets

The sets represented by the vertical line are ordinals. The first cardinal is $\emptyset$. The next one is 1, the next is 2, ... The first infinite cardinal is ω and is known as $\aleph_0$. $\aleph_1$ is the next cardinal after ω. It is not known where $\aleph_1$ falls between ω^ω and c, the cardinality of the reals. Note that $\omega + 1$, $\omega + \omega$, ω^2, and ω^ω are not cardinals.

The Continuum Hypothesis and Skolem's Paradox

Math 260B - Mathematical Logic

January 23, 1989

Definition: $Card(x) < Card(y)$ iff $Card(x) \leq Card(y)$ and $Card(y) \neq Card(x)$.

With the axiom of choice, $<$ is a linear order.

Definition: $Card(x)$ is the least ordinal α such that $Card(\alpha) = Card(x)$.

The following shows a partial list of the first few infinite cardinals.

$$\begin{aligned}
\aleph_0 &= \text{the first infinite cardinal} \\
\aleph_1 &= \text{the first uncountable cardinal} \\
&= \text{the first cardinal} > \omega \\
&\vdots \\
\aleph_{i+1} &= \text{the first cardinal} > \aleph_i \\
&\vdots \\
\aleph_\omega &= \bigcup_{i \in N} \aleph_i \\
&\vdots
\end{aligned}$$

Note that for every ordinal α, $\aleph_\alpha$ exists.

Definition: $c = Card({}^{\omega}2) = 2^{\omega}$, where ${}^{\omega}2$ is the set of maps from ω to 2 (remember, '2' denotes a set with 2 elements).

Continuum Hypotheses (CH): $c = \aleph_1$.

Generalized Continuum Hypotheses (GCH): For every infinite cardinal β, 2^β is the least cardinal $> \beta$.

Fact: CH is independent of ZFC; i.e., ZFC $\not\vdash$ CH (Cohen), and ZFC $\not\vdash \neg$ CH (Gödel).

Fact: GCH is also independent of ZFC (Cohen, Gödel).

Proposition: $Card(R) = c = 2^\omega$.

> *Claim 1:* $Card(R) \leq 2^\omega$.
>
> *Proof*: Let $h(r) \rightarrow f_r$ such that $f_r(i) = i^{\text{th}}$ digit of the binary expansion of r. Then each real r corresponds to a function f_r which defines a subset of the integers: $\{i : f(i) = 1\}$. Since there are 2^ω subsets of integers, $Card(R) \leq 2^\omega$. □
>
> *Claim 2:* $Card(R) \geq 2^\omega$.
>
> *Proof*: Code each subset of integers as a binary number as above. Now interpret each binary number as a ternary number. Not all of the reals will be represented by such numbers. Hence $Card(R) \geq 2^\omega$. □

Skolem's Paradox

- There is countable model, $\mathcal{A}$, of ZFC; i.e., $\mathcal{A} = (A, E)$, where A is countable, and $\mathcal{A} \models$ ZFC.
- $\mathcal{A} \models$ "there exists an uncountable set"; i.e., there is an $a \in A$ such that $\mathcal{A} \models$ "a is the power set of N and a is uncountable".
- $\mathcal{A} \models$ "the set of reals is an uncountable set"; i.e., there is an $a \in A$ such that $\mathcal{A} \models$ "$\forall x(x \subseteq N \leftrightarrow x \in a) \wedge \neg\exists$ 1-1 onto function $f : N \rightarrow a$".

The paradox is that A is countable so there are only countably many $b \in A$. The resolution of the paradox is that it is not the case that for every subset of N there is a corresponding object in A. Only the first order definable subsets are required to have corresponding objects in A, and there are only countably many of these. From our viewpoint outside of A, the a above contains only countably many elements (in the sense of E). And even though outside of A we can see a 1-1 onto $f : N \to a$, such an f can't be coded by a set in A.

Definition: $||\mathcal{L}|| = max\{Card(\mathcal{L}), \omega\}$; i.e., $||\mathcal{L}||$ is the size of $\mathcal{L}$.

Theorem: If Γ is a satisfiable set of sentences in the language $\mathcal{L}$, then Γ has a model of cardinality $\leq ||\mathcal{L}||$. (Note that we have proved this for countable $\mathcal{L}$.)

Theorem: [Löwenheim-Skolem] If Γ has an infinite model, then Γ has a model of cardinality α for all $\alpha \geq ||\mathcal{L}||$.

Proof: Let $\{c_\beta : \beta \in \alpha\}$ be a set of new constant symbols. (The β's are the ordinals in α. Note that the set of c_β's has cardinality α.) Let

$$\Pi = \Gamma \cup \{c_{\beta_1} \neq c_{\beta_2} : \beta_1 \neq \beta_2,\ \beta_1, \beta_2 < \alpha\}.$$

Every finite subset of Π is satisfiable since Γ has an infinite model. So by compactness, completeness, and our proof of completeness Π has a model of cardinality $\leq \alpha$ ($\alpha = ||\mathcal{L}(\Pi)||$); i.e., by compactness, Π is consistent, by completeness, Π has a model, and by our proof of completeness, Π has a model of cardinality $\leq \alpha$. Since the constants denote distinct objects, Π's model must have cardinality $= \alpha$. □

Skolem Functions and Herbrand's Theorem

Math 260B - Mathematical Logic

January 25, 1989

Definition: Given a first order prenex formula A, the skolemization of A, A^S, is a universal formula (only has universal quantifiers) defined as follows:

1. If A is universal or quantifier free, then $A^S = A$.
2. If $A = \forall x_1 \dots \forall x_k \exists y B(\vec{x}, y)$ with $k \geq 0$, then

   $$A^S = [\forall x_1 \dots \forall x_k B(\vec{x}, f_A(\vec{x}))]^S$$

   where f_A is a "new" k-ary function symbol. If $k = 0$, then $A^S = [B(c_A)]^S$ for a new constant symbol c_A.

The f_A's are called Skolem functions.

Theorem: A is satisfiable iff A^S is.

Definition: If Γ is a set of prenex formulas, then $\Gamma^S = \{A^S : A \in \Gamma\}$.

Theorem: [Skolem] Γ^S is satisfiable iff Γ is.
Proof: ($\Rightarrow$) Any model of Γ^S is also a model of Γ since $A^S \models A$. (Just ignore the extra function symbols.)
($\Leftarrow$) By compactness, it suffices to assume that Γ is finite. Let $\Gamma_0, \Gamma_1, \dots, \Gamma_k$ be sets of formulas where $\Gamma_0 = \Gamma$, $\Gamma_k = \Gamma^S$, and k is the number of existential quantifiers in Γ. Each Γ_i is one step in the process of skolemizing Γ; i.e., each Γ_{i+1} is obtained from Γ_i by replacing one $A_i = \forall\vec{x}\exists y B(\vec{x}, y)$ with $\forall\vec{x}B(\vec{x}, f_{A_i}(\vec{x}))$. Any model $\mathcal{A}_i$ of Γ_i can be expanded to a model $\mathcal{A}_{i+1}$ of Γ_{i+1} with $f_{A_i}^{\mathcal{A}_{i+1}}(\vec{a}) =$ some b such that $\mathcal{A}_i \models B(\vec{a}, b)$ for all $\vec{a} \in |\mathcal{A}_i|$. So if $\mathcal{A}_0 \models \Gamma$, then $\mathcal{A}_k \models \Gamma^S$. $\square$

Definition: If A is a prenex formula, then the Herbrandization of A, A^H, is defined by

1. If A is existential or quantifier free, then $A^H = A$.
2. If $A = \exists x_1 \ldots \exists x_k \forall y B(\vec{x}, y)$, then

$$A^H = [\exists x_1 \ldots \exists x_k B(\vec{x}, g_A(\vec{x}))]^H$$

   where g_A is a new k-ary function symbol. If $k = 0$, then $A^H = [B(c_A)]^H$ for a new constant symbol c_A.

Theorem: A^H is valid iff A is valid.

Proof: The dual of the previous proof (use negation). $\Box$

Theorem: Let Γ and A be in prenex normal form. Then $\Gamma \models A$ iff $\Gamma^S \models A^H$.

Proof: (outline) $\Gamma \models A$ iff $\Gamma \cup \{\neg A\}$ is not satisfiable. Let A^* be the prenex normal form of $\neg A$. Then $\Gamma \cup \{A^*\}$ is not satisfiable iff $\Gamma^S \cup \{(A^*)^S\}$ is not satisfiable iff $\Gamma^S \models \neg(A^*)^S$. After putting into prenex normal form and renaming f's to g's, $\neg(A^*)^S = A^H$. $\Box$

Herbrand's Theorem: Suppose that A is an existential sentence of the form $A = \exists x_1 \ldots \exists x_k B(x_1, \ldots, x_k)$ where B is quantifier free. Also suppose that Γ is a set of universal sentences and that $\Gamma \vdash A$. Finally, suppose that there is at least one constant symbol in the language. Then there are closed terms $t_{1,1}$, $t_{1,2}$, $\ldots$, $t_{1,k}$, $t_{2,1}$, $\ldots$, $t_{\ell,k}$, such that

$$\Gamma \vdash B(t_{1,1}, \ldots, t_{1,k}) \lor B(t_{2,1}, \ldots, t_{2,k}) \lor \ldots \lor B(t_{\ell,1}, \ldots, t_{\ell,k}).$$

Proof: Later.

Example: Let $A = \exists x Q(x)$, and let $\Gamma = \{Q(c) \lor Q(d)\}$. Then $\Gamma \vdash A$, and $\Gamma \vdash Q(c) \lor Q(d)$. This shows that ℓ may need to be greater than 1.

Example: Let $A = \forall x \exists y \forall z (P(x, z) \rightarrow P(x, y))$, and let $\Gamma = \emptyset$. $\models A$ since

$$\begin{aligned} A &\simeq \forall x \exists y (\exists x P(x, z) \rightarrow P(x, y)) \\ &\simeq \forall x (\exists x P(x, z) \rightarrow \exists y P(x, y)) \end{aligned}$$

$A^H = \exists y(P(c, g(y)) \to P(c, y))$. Now A^H is also valid by an earlier theorem. Since it is existential, we can apply Herbrand's theorem. The closed terms in the language of A^H are c, $g(c)$, $g(g(c))$, $\ldots$. So letting the B of Herbrand's theorem be $B(y) = P(c, g(y)) \to P(c, y)$, we get

$$\models [P(c, g(c)) \to P(c, c)] \vee [P(c, g(g(c))) \to P(c, g(c))].$$

(This is of the form $(D \to E) \vee (F \to D)$ which is a tautology.)

The general result is that given an arbitrary Γ and A, $\Gamma \vdash A$ iff $\Gamma^S \vdash A^H$. We can then apply Herbrand's theorem to get a concrete proof in the sense of having terms for which $\Gamma \vdash \exists \vec{x} B(\vec{x})$ is true. (Note that Herbrand's theorem doesn't tell us *how* to get the concrete proof, it only tells us that a concrete proof exists.)

Theorem: There is no recursive bound on the size of the terms $t_{i,j}$ where "size" is the number of function symbols in a term.

Proof: Otherwise, we could decide first order validity by testing whether or not the disjunction of all $B(t_1, \ldots, t_k)$'s with terms $t_1, \ldots, t_k$ whose size is less than or equal to the recursive bound is valid. $\Box$

Fact: There is no recursive bound on the ℓ of Herbrand's theorem.

Herbrand's Theorem, Hilbert-Style Proofs

Math 260B - Mathematical Logic

January 27, 1989

Herbrand's Theorem: (restated) Let Γ be a set of universal formulas, and let A be an existential formula of the form $\exists x_1 \ldots \exists x_k B(x_1, \ldots, x_k)$. The language of Γ and A is L, and L must have at least one constant symbol. If $\Gamma \models A$, then there are terms $t_{i,j}$ with $i = 1 \ldots \ell$, and $j = 1 \ldots k$ such that

$$\Gamma \vdash B(t_{1,1}, \ldots t_{1,k}) \vee \ldots \vee B(t_{\ell,1}, \ldots t_{\ell,k}).$$

Definition: Let A be a quantifier free sentence. A is a *quasi-tautology* iff there is a (finite) set of instances of the equality axioms which imply A ("finite" is unnecessary by the compactness theorem.)

For example, $c = d \rightarrow d = c$ is a quasi-tautology but not a tautology. Why? Because if we treat each atomic formula as a propositional variable, then we have $A \rightarrow B$, which is not a tautology. But the equality axiom $\forall x \forall y(x = y \rightarrow y = x)$ implies $A \rightarrow B$.

Definition: $\{A_1, \ldots, A_k\}$ is *quasi-tautologically inconsistent* iff $\neg(A_1 \wedge \ldots \wedge A_k)$ is a quasi-tautology. Equivalently, $\{A_1, \ldots, A_k\}$ are quasi-tautologically inconsistent iff there is a (finite) set of instances of the equality axioms which together with $A_1, \ldots A_k$ are tautologically inconsistent.

Main Lemma: (for proving Herbrand's theorem) If Γ is an unsatisfiable set of universal sentences, then there is some finite set of instances of formulas in Γ which is quasi-tautologically inconsistent (or which, together with some finite set of equality axioms, are tautologically inconsistent).

Proof: Since Γ is unsatisfiable, it has a refutation R. The finite set of instances of formulas of Γ in R is quasi-tautologically inconsistent. $\square$

Note that "instance of $\forall x_1 \ldots \forall x_k B(x_1, \ldots, x_k)$" means any formula of the form $B(t_1, \ldots, t_k)$ where B is quantifier free and the t_i's are closed terms. These are also called *ground* instances.

Proof of Herbrand's theorem: Since $\Gamma \vdash A$, there is a refutation R of $\Gamma \cup \{\forall x_1 \ldots \forall x_k \neg B(x_1, \ldots, x_k)\}$. In this refutation, there is a finite set of instances of $\forall \vec{x} \neg B(\vec{x})$:

$$\{\neg B(t_{1,1}, \ldots, t_{1,k}), \neg B(t_{2,1}, \ldots, t_{2,k}), \ldots, \neg B(t_{\ell,1}, \ldots, t_{\ell,k})\}.$$

Claim:

$$\Gamma \vdash B(t_{1,1}, \ldots, t_{1,k}) \vee \ldots \vee B(t_{\ell,1}, \ldots, t_{\ell,k}).$$

$\Gamma \cup \{\neg(B(t_{1,1}, \ldots, t_{1,k}) \vee \ldots \vee B(t_{\ell,1}, \ldots, t_{\ell,k}))\}$ has a refutation R'. R' is constructed from R by deleting the $\neg B(t_{i,1}, \ldots, t_{i,k})$'s and adding $\neg(B(\vec{t_1}) \vee \ldots \vee B(\vec{t_\ell}))$. □

(Note: the $\neg B(t_{i,1}, \ldots, t_{i,k})$'s are every formula of the form

$$\forall x_p \forall x_{p+1} \ldots \forall x_k \neg B(s_1, \ldots, s_{p-1}, x_p, \ldots, x_k)$$

obtained by applications of UI to $\forall \vec{x} \neg B(\vec{x})$. Also note that $\neg(B(\vec{t_1}) \vee \ldots \vee B(\vec{t_\ell}))$ is tautologically equivalent to $\neg B(\vec{t_1}) \wedge \neg B(\vec{t_2}) \wedge \ldots \wedge \neg B(\vec{t_\ell})$; so, intuitively, if $\neg B(\vec{t_1}), \ldots, \neg B(\vec{t_k})$ are quasi-tautologically inconsistent, then so is $\neg(B(\vec{t_1}) \vee \ldots \vee B(\vec{t_k}))$.)

Hilbert-Style Proofs

An alternative proof system to the refutation system we have used is a Hilbert-style proof system for obtaining valid first order formulas. The axioms of such a proof system are:

1. Every tautology is an axiom; i.e. $\forall x \phi \rightarrow \forall x \phi$.
2. $\forall x \phi \rightarrow \phi(t/x)$ for every ϕ, x, and t.
3. $\phi(t/x) \rightarrow \exists x \phi$.
4. Every equality axiom.
5. If ϕ is an axiom according to (1) - (4), then $\forall x_1 \ldots \forall x_k \phi$ is also an axiom.

There is only one rule for deriving new formulas called *modus ponens*:

$$\frac{\phi \to \psi \qquad\quad \phi}{\psi}$$

A proof is a sequence of formulas $A_1, \ldots, A_r$ such that each A_i is either an axiom or is inferred by modus ponens from two earlier formulas.

Completeness and Soundness: A is valid iff A has a proof.

Hilbert-style proofs may be much shorter than refutations of $\neg A$, but can be harder to find. A Hilbert-style system also makes it more difficult to prove Herbrand's theorem (and the Craig interpolation theorem).

Fact: For arbitrarily large n, there is a valid formula A with a Hilbert-style proof of n lines which requires $2 \Uparrow cn$ lines for a refutation of $\neg A$ where c is some "reasonable" constant.

Applications of Herbrand's Theorem, Resolution

Math 260B - Mathematical Logic

February 1, 1989

As an application of Herbrand's theorem, consider the following theorem from algebra.

Theorem: Suppose that $\cdot$ is a binary associative operation, and suppose that equations $a \cdot x = b$ and $y \cdot a = b$ always have solutions x and y. Then there is a right identity element.

Proof: (intuitive) Pick an arbitrary element a. Let u be a solution to $a \cdot u = a$.

> *Claim:* For all b, $b \cdot u = b$.
>
> *Proof*: Let c be such that $c \cdot a = b$. Then
>
> $$c \cdot (a \cdot u) = c \cdot a = b,$$
>
> and also
>
> $$c \cdot (a \cdot u) = (c \cdot a) \cdot u = b \cdot u.$$

This completes the intuitive proof of the claim and the theorem. □

Now we'll recast this theorem and its proof in first order logic. The suppositions are captured by the following three properties:

$$\begin{array}{rcl} \Gamma & = & \{ \;\; \forall x \forall y \forall z((x \cdot y) \cdot z = x \cdot (y \cdot z)), \\ & & \quad\; \forall x \forall z \exists y (x \cdot y = z), \\ & & \quad\; \forall y \forall z \exists x (x \cdot y = z) \;\; \}, \end{array}$$

and the conclusion is $A = \exists u \forall x (x \cdot u = x)$.

Theorem: $\Gamma \vdash A$ (or $\Gamma \cup \{B\}$ is inconsistent, where $B = \forall u \exists x (x \cdot u \neq x)$).

Proof:

$$\begin{array}{rcll} \Gamma^S & = & \{ & \forall x \forall y \forall z((x \cdot y) \cdot z = x \cdot (y \cdot z)), \quad (i) \\ & & & \forall x \forall z(x \cdot f(x,z) = z), \quad (ii) \\ & & & \forall y \forall z(g(y,z) \cdot y = z) \ \} \quad (iii) \end{array}$$

$$\begin{array}{rcl} B^S & = & \forall u(h(u) \cdot u \neq h(u)), \text{ and} \\ A^H & = & \exists u(h(u) \cdot u = h(u)). \end{array}$$

The theorem says that $\Gamma^S \cup \{B^S\}$ is inconsistent or equivalently that $\Gamma^S \vdash A^H$. A refutation of $\Gamma^S \cup \{B^S\}$ contains:

1. $a \cdot f(a,a) = a$ by 2 applications of UI to *ii*,
2. $h(f(a,a)) \cdot f(a,a) \neq h(f(a,a))$ by UI to B^S,
3. $g(a, h(f(a,a))) \cdot a = h(f(a,a))$ by 2 applications of UI to *iii*, and
4. $(g(a,h(f(a,a))) \cdot a) \cdot f(a,a) = g(a,h(f(a,a))) \cdot (a \cdot f(a,a))$ by 3 applications of UI to *i*.

Now, (1) - (4) are quasi-tautologically inconsistent.

> *Homework:* find instances of the equality axioms which make (1) - (4) tautologically inconsistent.

In fact, we have shown that

$$\Gamma^S \vdash h(f(a,a)) \cdot f(a,a) = h(f(a,a)).$$

(Recall that $\Gamma^S \vdash \exists u(h(u) \cdot u = h(u))$ implies by Herbrand's theorem that

$$\Gamma^S \vdash h(t_1) \cdot t_1 = h(t_1) \vee \ldots \vee h(t_k) \cdot t_k = h(t_k).$$

In the above usage, $k = 1$, and $t_1 = f(a,a)$.)
Furthermore, $\Gamma^S \vdash \forall x(x \cdot f(a,a) = x)$.
Proof: Homework. □

Resolution

First we want a refutation proof system for propositional logic. (We don't want a theorem prover to use the method of truth tables because that is guaranteed to take exponential time. We can't *always* do better than exponential time with our current state of knowledge, but we would like a theorem prover that does better often in practice.) Assume that formulas are in conjunctive normal form. (This works well in practice, but note that some formulas grow exponentially in size when converted to CNF.)

Definition: A *literal* is either a variable P_i or the negation of a variable $\neg P_i$.

Definition: The *complement* of P_i is $\neg P_i$ and vice versa.

Definition: A *clause* is a finite set of literals. (Think of a clause as the disjunction of its literals.)

Definition: If τ is a truth assignment and C is a clause, then τ satisfies C ($\tau \models C$) iff $\tau(X) =$ true for some literal $X \in C$. (Note that if C is the empty clause, then no τ makes C true.)

Definition: If S is a set of clauses, then $\tau \models S$ iff $\tau \models C$ for every $C \in S$. Note that if $S = \emptyset$, then any τ makes S true.)

A propositional formula is expressed as a set of clauses. For example $(P_1 \vee P_2) \wedge (\neg P_3 \vee P_4)$ is represented by $\{\{P_1, P_2\}, \{\neg P_3, P_4\}\}$.

Definition: Let C_1, and C_2 be clauses with $P_i \in C_1$ and $\neg P_i \in C_2$. The *resolvent* of C_1 and C_2 with respect to P_i is the clause

$$(C_1 \setminus \{P_i\}) \cup (C_2 \setminus \{\neg P_i\}).$$

Proposition: If $\tau \models C_1$ and $\tau \models C_2$, then τ satisfies any resolvent of C_1 and C_2.

Proof: Easy. $\square$

Resolution Refutations

Math 260B - Mathematical Logic

February 3, 1989

Definition: Let Γ be a set of clauses. A *resolution refutation* from Γ is a sequence of clauses $C_1, \ldots, C_t$ such that each C_i is either in Γ or is the resolvent of two earlier clauses C_j and C_k with $j, k < i$ and such that $C_t = \emptyset$, the empty clause.

Theorem: Γ has a resolution refutation iff Γ is unsatisfiable.

Proof: ($\Rightarrow$ (soundness)) If Γ is satisfiable by a truth assignment σ, then by the previous proposition, $\sigma \models C_i$ for $i = 1, \ldots, t$. But $\sigma \not\models \emptyset$. So Γ has no resolution refutation.

($\Leftarrow$ (completeness)) Let Γ^* be the set of clauses which can be obtained from Γ by resolution. By definition, Γ^* contains the empty clause iff Γ has a resolution refutation.

> *Remark:* Γ^* is finite if Γ is finite.
>
> *Proof*: Finite Γ implies that there are only finitely many literals in clauses in Γ. Any clause in Γ^* contains only these literals. Since every clause in Γ is finite, there are only finitely many resolvents, and hence only finitely many clauses in Γ^*. $\Box$

Suppose that the empty set is not in Γ^* so that Γ has no resolution refutation. We want to find a σ such that $\sigma \models \Gamma$. Define σ by setting $\sigma(P_1)$, $\sigma(P_2), \ldots$ sequentially as follows:

$$\sigma(P_i) = \begin{cases} \text{true} & \text{if this doesn't force any clause} \\ & \text{in } \Gamma^* \text{ to be false} \\ \text{false} & \text{otherwise} \end{cases}$$

I.e., set $\sigma(P_i)$ to true if there is no clause containing only literals among

$$P_1, \neg P_1, P_2, \neg P_2, \ldots, P_i, \neg P_i$$

for which every literal is given the value false; otherwise, set $\sigma(P_i)$ to false.

> *Claim:* $\sigma \models \Gamma^*$.
>
> *Proof*: Assume towards a contradiction that there is a clause $C \in \Gamma^*$ such that $\sigma \not\models C$. Pick i to be the minimum i such that the values of σ for $P_1, \ldots, P_i$ force C to be false. From the definition of σ, it follows that $\sigma(P_i) =$ false and that there is a clause $C^{'}$ in Γ such that it would have been forced to be false if we had set $\sigma(P_i) =$ true. Let D be the resolvent of C and $C^{'}$ with respect to P_i. Note that $D \in \Gamma^*$. D was already forced to be false by values of σ on $P_1, \ldots, P_{i-1}$. $P_i \in C$ and $\neg P_i \in C^{'}$ since $\sigma(P_i) =$ false. So resolving C and $C^{'}$ with respect to P_i makes sense. Since $D = (C \setminus \{P_i\}) \cup (C^{'} \setminus \{\neg P\})$, and P_i can't be in $C^{'}$, $P_i \notin D$. Similarly for $\neg P_i$. Every other P_j or $\neg P_j$ in D has $j < i$ and is set false by σ. But this contradicts our choice of i as being the minimum.

This completes the proof of the claim and the completeness part of the theorem. $\Box$

Example: of the contrdiction obtained above. Suppose that $i = 4$, $C = \{P_1, \neg P_3, P_4\}$ and $C^{'} = \{\neg P_2, \neg P_3, \neg P_4\}$. Then

$$\begin{aligned} \sigma(P_1) &= \text{false} \\ \sigma(P_2) &= \text{true} \\ \sigma(P_3) &= \text{true} \\ \sigma(P_4) &= \text{false.} \end{aligned}$$

So $D = \{P_1, \neg P_2, \neg P_3\}$ which is already set false by σ on P_1, P_2, and P_3.

Example: (of a resolution refutation)

$$\{A \to B \vee C,\ B \to D,\ C \to D\} \models A \to D$$

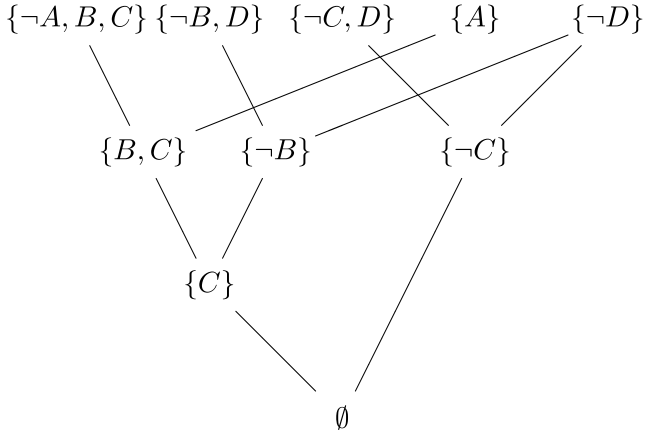
iff

$$\{\neg A \vee B \vee C,\ \neg B \vee D,\ \neg C \vee D,\ A \wedge \neg D\}$$

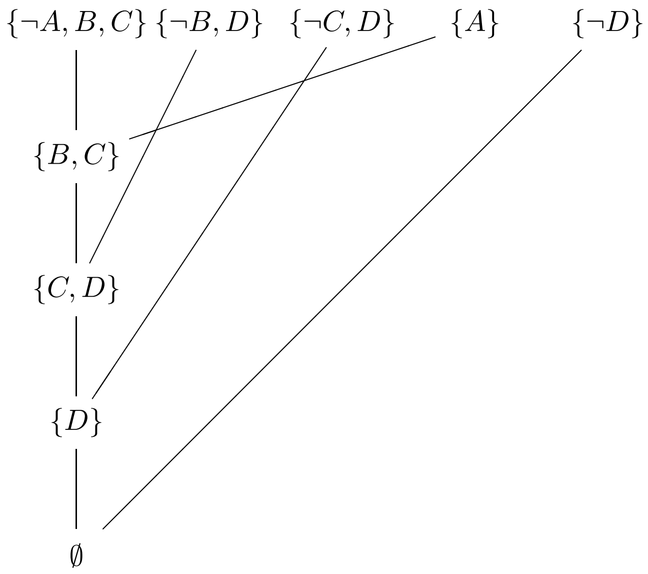
is unsatisfiable iff the set of clauses

$$\{\{\neg A, B, C\},\ \{\neg B, D\},\ \{\neg C, D\},\ \{A\}, \{\neg D\}\}$$

is unsatisfiable (has a resolution refutation). The following picture illustrates the resolution refutation:



This is an example of what is called *unit* resolution since in each step, one of the resolvands is a unit clause; i.e., it has a single literal. There are other ways to resolve. The following is a picture of *linear* resolution:



Linear resolution always resolves with the result of the most recent resolvent. A third method is called *input* resolution and always resolves with an "input" clause; i.e., one of the original clauses.

Fact: Linear resolution is complete.

Fact: Unit and input resolution are not complete.

Definition: A clause C is a *Horn clause* iff it contains at most one unnegated variable.

Example: The following are Horn clauses:

$$\{P_i, \neg P_{j_1}, \neg P_{j_2}, \ldots, \neg P_{j_k}\}, \text{ and}$$
$$\{\neg P_{j_1}, \neg P_{j_2}, \ldots, \neg P_{j_k}\}$$

The intuitive meaning of the first Horn clause is

$$P_{j_1} \wedge P_{j_2} \wedge \ldots \wedge P_{j_k} \rightarrow P_i.$$

Fact: Unit and input resolution are complete for Horn clauses.

Horn clauses work well for expressing facts in the real world. They are also used for expressing facts in Prolog. But some facts can't be expressed with Horn clauses.

Fact: $A \vee B$ can not be expressed by the conjunction of a set of Horn clauses.

First order Horn clauses are more expressive. A remarkable property is the proof of the fact that Horn clauses can express the statement "there are not exactly p objects" for p a prime number. For example, if $p = 7$, then in first order logic, we would say

$$\exists x_1 \ldots \exists x_6 \forall y (\bigvee_{i=1}^{6} y = x_i) \ \vee \ \exists x_1 \ldots \exists x_8 (\bigwedge_{1 \leq i < j \leq 8} x_i \neq x_j).$$

I.e., there are fewer than 7 objects or there are more than 7 objects. There is a general theorem that implies that this kind of statement can be expressed in terms of a conjunction of Horn sentences; the proof of this theorem seems to depend on the continuum hypothesis! However explicit examples are known of Horn clauses that express "there are not exactly p objects" for p prime.

Unification

Math 260B - Mathematical Logic

February 6, 1989

We would like to lift the technique of resolution refutation from propositional logic to first order logic. We'll use Herbrand's theorem and Skolem functions to do this. The general problem is to find a refutation of a set of universal formulas: $\forall x \forall y \phi$, $\forall x^{'} \forall y^{'} \phi^{'} \ldots$.

If there is a refutation, then by Herbrand's theorem, there is a set of instantiations of these universal formulas which are tautologically inconsistent. If the quantifier free parts $\phi, \phi^{'}, \ldots$ are in conjunctive normal form, then propositional resolution will give a refutation if one exists. So there are two parts to this process: picking the terms of Herbrand's theorem and finding the refutation. The potential problems are that the process of finding the terms may not terminate and that finding the refutation could take exponential time in the number of terms needed for instantiations.

Unification will provide us with a method for automatically choosing the terms with which to instantiate as we generate a resolution refutation. This allows us to avoid having to know the terms ahead of time.

Example: As an informal example of what we mean by unification consider the following two sentences:

$$\begin{array}{l} \forall x \text{Likes}(x, \text{Mother}(x)) \\ \forall x \forall y (\text{Likes}(x, y) \rightarrow \text{Know}(x, y)) \end{array}$$

These two sentences imply that everyone knows their mother. To prove this, the process of unification attempts to match the terms Likes$(x, \text{Mother}(x))$ and Likes(x, y) to come up with:

$$\begin{array}{l} \forall x \text{Likes}(x, \text{Mother}(x)) \\ \forall x (\text{Likes}(x, \text{Mother}(x)) \rightarrow \text{Know}(x, \text{Mother}(x))) \end{array}$$

With this and modus ponens, we can infer

$$\forall x(\text{Know}(x, \text{Mother}(x))$$

(Note: in automatic theorem proving, we sometimes omit the outer $\forall$'s and implicitly assume that free variables are universally quantified.)

Definition: A *substitution* σ, is a mapping whose domain is a finite set of variables and whose range is a set of terms (open or closed). σ is denoted by $(t_1/x_1, \ldots, t_k/x_k)$, where $\sigma(x_j) = t_j$.

Definition: If A is a quantifier free formula, then $A\sigma$ is the formula obtained by replacing every (free) occurrence of x_i in A by t_i for $i = 1, \ldots, k$ simultaneously. (Note that we specified that A is quantifier free. This is because we'll never need this rule for anything but quantifier free formulas, and we don't want to have to deal with the problems of clashing variables.)

Definition: If $A_1, \ldots, A_k$ are quantifier free formulas, then σ *unifies* $\{A_1, \ldots, A_k\}$ iff $A_1\sigma = A_2\sigma = \ldots = A_k\sigma$. In this case, we say that $\{A_1, \ldots, A_k\}$ are *unifiable*.

Definition: If $\sigma = (t_1/x_1,\ \ldots,\ t_k/x_k)$ and $\tau = (r_1/x_{i_1},\ r_2/x_{i_2},\ \ldots,\ r_n/x_{i_n},\ s_1/y_1,\ \ldots,\ s_m/y_m)$, then the *composition* of σ and τ is $\sigma\tau = (t_1\tau/x_1,\ \ldots,\ t_k\tau/x_k,\ s_1/y_1,\ \ldots,\ s_m/y_m)$. ($\tau$ is a substitution with some number of x's and some new y's in its domain; i.e., $0 \leq n \leq k$ and $0 \leq m$.)

Definition: σ is a *most general unifier* (mgu) of $\{A_1, \ldots, A_k\}$ iff

1. σ unifies $\{A_1, \ldots, A_k\}$, and
2. any other unifier λ of $\{A_1, \ldots, A_k\}$ can be expressed as $\lambda = \sigma\tau$ for some substitution τ.

Proposition: $A(\sigma\tau) = (A\sigma)\tau$.

Proof: Clear after inspection. $\Box$

Example: Suppose we want to unify $\{P(f(a), b),\ P(c, h(c))\}$ where a, b, and c are variables. Then we want $c \mapsto f(a)$ and $b \mapsto h(f(a))$. So $\sigma = (f(a)/c,\ h(f(a))/b)$. To check that σ is a unifier we compute

$$P(f(a), b)\sigma = P(f(a), h(f(a)))$$

and

$$P(c, h(c))\sigma = P(f(a), h(f(a))).$$

It turns out that σ is also an mgu. Another mgu is $\sigma^{'} = (d/a,\ h(f(d))/b,\ f(d)/c)$ obtained from σ by changing a to d. As an example of a unifier λ such that $\lambda = \sigma\tau$ for some substitution τ take $\lambda = (h(f(f(a)))/b,\ f(f(a))/c)$ and $\tau = (f(a)/a)$.

Proposition: If σ unifies $\{A_1, \ldots, A_k\}$ then so does $\sigma\tau$ for any τ.

Proof: Easy. □

Proposition: If σ is an mgu for $A_1, \ldots, A_k$, then the unifiers of $A_1, \ldots, A_k$ are precisely the substitutions $\sigma\tau$ for any τ.

Note that unifiers don't always exist for a set of sentences. As examples, none of the following three sets of sentences are unifiable:

$$\begin{array}{l}\{P(h(x)),\ P(g(x))\}\\ \{P(x),\ Q(x)\}\\ \{P(f(x)),\ P(x)\}\end{array}$$

The first is not unifiable because we can't match g and h, the second because P and Q are different, and the third because x and $f(x)$ can't be matched.

The following definition is needed in order to develop a method for finding mgu's.

Definition: Let $\{A_1, \ldots, A_k\}$ be a set of atomic formulas starting with a common predicate symbol. Let i be the smallest positive integer such that some A_j and $A_{j'}$ disagree on their i^{th} symbol. Then the *disagreement set* of $\{A_1, \ldots, A_k\}$ is the set of terms $\{d_1, \ldots d_k\}$ where d_j is the term starting at the i^{th} symbol in A_j.

Example: The disagreement set of

$$\begin{aligned} \{ \quad & P(h(x,y)), \\ & P(h(f(y),g(z))), \\ & P(h(f(g(z)),y)) \quad \} \end{aligned}$$

is $\{x, f(y), f(g(z))\}$. In this case $i = 5$.

Some things to note about disagreement sets:

1. The cardinality of a disagreement set ≥ 2.
2. If $A_1, \ldots A_k$ are unifiable with unifier σ, then $d_1\sigma = \ldots = d_k\sigma$. It follows that there is a function symbol f such that each d_i either begins with f or is a variable. Also, at least one d_i is a variable.

Unification Algorithm

Input: Atomic formulas $A_1, \ldots, A_k$ with a common predicate symbol.

Output: Either a substitution σ_A, or the phrase "not unifiable".

Steps

1. σ_0 = identity substitution. If $A_1, \ldots, A_k$ are all equal, set $\sigma_A = \sigma_0$ and halt.
2. Loop with $\ell = 1, 2, \ldots$
   (a) Compute the disagreement set $\{d_1, \ldots, d_k\}$ of $\{A_1\sigma_{\ell-1}, \ldots, A_k\sigma_{\ell-1}\}$.
   (b) If no d_i is a variable, then output "not unifiable" and halt.
   (c) Let d_i be a variable, and let d_j be some other member of the disagreement set such that $d_i \neq d_j$.
   (d) If the variable d_i occurs in d_j, then output "not unifiable" and halt. (This step is commonly referred as the "occurs check"; in some Prolog implementations, it may be turned off for the sake of speed.)
   (e) Let σ_ℓ be $\sigma_{\ell-1}(d_j/d_i)$.
   (f) If σ_ℓ unifies $\{A_1, \ldots, A_k\}$, then set $\sigma_A = \sigma_\ell$ and halt; otherwise continue loop.

Unification Theorem, Robinson Resolution

Math 260B - Mathematical Logic

February 8, 1989

Last time, we presented the unification algorithm. To summarize, the algorithm takes as input a set of quantifier free formulas, $\{A_1, \ldots, A_k\}$, and computes a sequence of substititions starting from σ_0, the identity substitution. σ_ℓ was constructed by composing $\sigma_{\ell-1}$ with (d_j/d_i) where d_i was a variable, d_j was a term, and both d_i and d_j were in the disagreement set $\{A_1\sigma_{\ell-1}, \ldots, A_k\sigma_{\ell-1}\}$. The algorithm either halted with an mgu of $\{A_1, \ldots, A_k\}$ or halted and printed "not unifiable".

Note that the mgu output by the algorithm, if it exists, is not unique. For example, the following mgu's all unify the set $\{P(x,y), P(y,x)\}$:

$$\begin{aligned}\sigma_1 &= (y/x)\\ \sigma_2 &= (x/y)\\ \sigma_3 &= (z/x, z/y)\end{aligned}$$

The algorithm doesn't specify which substitution to make. But note also that any two mgu's of a set of quantifier free sentences are identical up to renaming of variables.

Proposition: The unification algorithm always halts.

Proof: $\{A_1\sigma_\ell, \ldots, A_k\sigma_\ell\}$ has one less variable than $\{A_1\sigma_{\ell-1}, \ldots, A_k\sigma_{\ell-1}\}$ has. (Because of the occurs check, d_i will not appear in $\{A_1\sigma_\ell, \ldots, A_k\sigma_\ell\}$.) □

Unification Theorem: $\{A_1\sigma_\ell, \ldots, A_k\sigma_\ell\}$ has a unifier iff the unification algorithm produces an mgu.

Proof: Suppose that λ unifies $A_1, \ldots, A_k$. To prove the theorem, it suffices to show by induction on ℓ that the unification algorithm doesn't halt with failure and that there is a τ_ℓ such that $\lambda = \sigma_\ell\tau_\ell$.

Basis: $\ell = 0$. Then σ_0 is the identity substitution, and $\tau_0 = \lambda$.

Induction: Suppose that $\lambda = \sigma_{\ell-1}\tau_{\ell-1}$. Let $\{d_1, \ldots, d_k\}$ be the disagreement set of $\{A_1\sigma_{\ell-1}, \ldots, A_k\sigma_{\ell-1}\}$. Now $\tau_{\ell-1}$ unifies $d_1, \ldots, d_k$. One of the d's is a variable, hence the unification algorithm doesn't fail at this point. Let d_i be the variable picked by the algorithm, and let d_j be the other term picked. So $\sigma_\ell = \sigma_{\ell-1}(d_j/d_i)$. Furthermore, the occurs check is satisfied because $\{d_1, \ldots, d_k\}$ is unifiable. Now $d_i\tau_{\ell-1} = d_j\tau_{\ell-1}$. So $\tau_{\ell-1}$ includes $(d_j\tau_{\ell-1}/d_i)$. Let τ^* be $\tau_{\ell-1} \setminus \{(d_j\tau_{\ell-1}/d_i)\}$.

> *Claim:* $\sigma_{\ell-1}(d_j/d_i)\tau^* = \sigma_{\ell-1}\tau_{\ell-1}$.
>
> *Proof*: Actually, $(d_j/d_i)\tau^* = \tau_{\ell-1}$ since $d_j\tau^* = d_j\tau_{\ell-1}$ since d_i does not occur in d_j. □

So take $\tau_\ell = \tau^*$. Then $\sigma_\ell = \sigma_{\ell-1}(d_j/d_i)$, and $\sigma_\ell\tau_\ell = \sigma_{\ell-1}\tau_{\ell-1} = \lambda$ by the claim. □

Example: The unification of $\{P(x, h(y, y)), P(h(z, z), z)\}$ goes as follows:

- $\sigma_0 =$ identity.
- The first disagreement set is $\{x, h(z, z)\}$.
- $\sigma_1 = \sigma_0(h(z, z)/x) = (h(z, z)/x)$.
- The disagreement set of

  $$\{P(x, h(y, y))\sigma_1, P(h(z, z), z)\sigma_1\}$$

  is $\{h(y, y), z\}$.
- $\sigma_2 = \sigma_1(h(y, y)/z) = (h(h(y, y), h(y, y))/x, h(y, y)/z)$.
- Since the next disagreement set is empty, σ_2 is an mgu.

As a check of σ_2,

$$P(x, h(y, y))\sigma_2 = P(h(z, z), z)\sigma_2 = P(h(h(y, y), h(y, y)), h(y, y)).$$

Robinson Resolution

The reason for unification is so that we can lift propositional resolution to resolution of universal sentences.

Definition: Let $C = (Q_1x_1)(Q_2x_2)\dots(Q_kx_k)C^M$, where the Q's are quantifiers, and C^M is quantifier free. Then C^M is called the *matrix* of C.

Proposition: Every first order universal formula is logically equivalent to a conjunction of universal formulas whose matrices are disjunctions of atomic and negated atomic formulas.

Proof: Let A be a first order universal formula. Without loss of generality, let A be $\forall\vec{x}A^M$. Express A^M in conjunctive normal form as

$$A^M \simeq \bigwedge_{i=1}^{n}\bigvee_{j=1}^{m_i} B_{i,j},$$

where each $B_{i,j}$ is either an atomic or negated atomic formula. Now

$$A \simeq \bigwedge_{i=1}^{n}(\forall\vec{x}\bigvee_{j=1}^{m_i} B_{i,j}). \ \square$$

So with A as above, we can associate clauses of the form $\{B_{i,h} : j = 1, \dots, m\}$ for $i = 1, \dots, n$.

The following definitions are analogs of definitions for propositional resolution.

Definition: A *literal* is an atomic or negated atomic formula. A *ground literal* is one with no variables.

Definition: A *clause* is a finite set of literals. A *ground clause* is a finite set of ground literals.

Definition: The meaning of a clause C is denoted by $FO(C)$ (FO stands for first order).

$$FO(C) = \forall\vec{x}(\bigvee_{B\in C} B),$$

where $\vec{x}$ includes all variables used in members of C.

Definition: A clause C is true in $\mathcal{A}$ iff $\mathcal{A} \models FO(C)$. A set of clauses S is true in $\mathcal{A}$ iff $\mathcal{A} \models FO(C)$ for all $C \in S$.

Proposition: Let Γ be a set of universal sentences. Then there is a set of clauses which is logically equivalent to Γ.

Proof: Use the above construction on every formula in Γ. $\Box$

Definition: [Robinson resolution] Let C and C^* be clauses. Let D be obtained from C^* by renaming variables so that C and D have no variables in common. Let $C^{'} \subseteq C$ be such that every member of $C^{'}$ is of the form $P(\vec{t})$ for some predicate symbol P. Similarly, let $D^{'} \subseteq D$ be such that every member of $D^{'}$ is of the form $\neg P(\vec{t})$ for the same predicate symbol P. Let σ be an mgu of $C^{'} \cup \{P(\vec{t}) : \neg P(\vec{t}) \in D^{'}\}$. If σ exists, then $(C\sigma \setminus C^{'}\sigma) \cup (D\sigma \setminus D^{'}\sigma)$ is a *resolvent* of C and C^*.

This resolving can be thought of as consisting of two steps:

1. "Factoring:" form $C\sigma$ from C and $D\sigma$ from D.
2. "Resolving:" resolve $C\sigma$ and $D\sigma$ with respect to $C^{'}\sigma$ in the propositional sense.

Example: Suppose we want to resolve

$$\begin{aligned} C &= \{\text{Loves}(x, \text{Mother}(x))\}, \text{ and} \\ C^* &= \{\neg\text{Loves}(x, y), \text{Know}(x, y)\}. \end{aligned}$$

Renaming common variables in C^*, we get

$$D = \{\neg\text{Loves}(z, y), \text{Know}(z, y)\}.$$

Now let $C^{'} = C$, and let $D^{'} = \{\neg\text{Loves}(z, y)\}$. An mgu of

$$\{\text{Loves}(x, \text{Mother}(x)), \neg\text{Loves}(z, y)\}$$

is $\sigma = (x/z, \text{Mother}(x)/y)$. Applying σ to C and D gives

$$\begin{aligned} C\sigma &= \{\text{Loves}(x, \text{Mother}(x))\}, \text{ and} \\ D\sigma &= \{\neg\text{Loves}(x, \text{Mother}(x)), \text{Know}(x, \text{Mother}(x))\}. \end{aligned}$$

From these, we can propositionally resolve to get $\{\text{Know}(x, \text{Mother}(x))\}$.

Robinson Resolution, Resolution Theorem, Prolog

Math 260B - Mathematical Logic

February 10, 1989

A modified definition of resolution: σ is an mgu that the unification algorithm produces. The domain of σ is a subset of the variables in $C^{'}$ and $D^{'}$. The range of σ consists of terms involving only variables that appear in $C^{'}$ and $D^{'}$. For instance, suppose that $C = \{P(x),\ Q(y)\}$, and $D = \{\neg Q(y),\ R(y)\}$. Then $C^{'} = \{Q(y)\}$, and $D^{'} = \{\neg Q(y)\}$, and we want C and D to resolve to $\{P(x),\ R(y)\}$ not $\{P(x),\ R(x)\}$. The first resolvent is equivalent to $\forall x \forall y (P(x) \vee R(y))$ which is a more general resolution than the second resolvent which is equivalent to $\forall x (P(x) \vee R(x))$.

Example: This example is a case where resolving with respect to a single literal doesn't work. Let $C = \{P(x),\ P(a)\}$, and let $C^{*} = \{\neg P(x),\ \neg P(a)\}$, where a is a constant and x is a variable. Then $D = \{\neg P(y),\ \neg P(a)\}$, and $C^{'} = C$ and $D^{'} = D$. An mgu is $\sigma = (a/x,\ a/y)$, $C^{'}\sigma = \{P(a)\}$, $D^{'}\sigma = \{\neg P(a)\}$, and the resolvent is $\emptyset$.

The modification of Robinson Resolution is that we've dropped the "factoring" part; i.e. $C^{'}$ and $D^{'}$ don't have to be singletons. As the above example shows, we can't always resolve to get $\emptyset$ (even with a sequence of resolutions) even though the input clauses may be inconsistent. Without this modification, we could only get $\{P(x),\ \neg P(a)\}$ and $\{P(x),\ \neg P(y)\}$ as resolvents.

The explicit details of factoring are to take C and $C^{'} \subseteq C$ and infer $C\sigma$ where σ is an mgu of $C^{'}$. We do this to unify $C^{'}$ itself and $D^{'}$ itself, and then we unify these 2 singletons. (In all, we use three mgu's.)

Resolution Theorem: Let Γ be a set of universal sentences with matrices which are disjunctions of literals. (Note that Γ can be the equivalent of any set of first order sentences.) Let S be the equivalent set of clauses. Γ is unsatisfiable iff S has a Robinson resolution refutation.

Proof: ($\Leftarrow$ (soundness)) Let $C_1, \ldots, C_t$ be a refutation from S ($C_t = \emptyset$).

> *Claim:* If $\mathcal{A} \models \Gamma$, then $\mathcal{A} \models FO(C_i)$, for $i = 1, \ldots, t$. (Remember, $FO(C_i) = \forall \vec{x} \bigvee_{B \in C_i} B$.)
>
> *Proof*: (by induction on i)
>
> *Basis*: $C_i \in S$. Then since $\mathcal{A} \models \Gamma$, $\mathcal{A} \models FO(C_i)$ for each $C_i \in S$.
>
> *Induction*: C_i is inferred from C_j and C_k for $j, k < i$. So $\mathcal{A} \models FO(C_j) \land FO(C_k)$. Now let $C_j^{'} \subseteq C_j$, and $C_k^{'} \subseteq C_k$. For convenience, assume that C_j and C_k have no variables in common; if not, then just rename common variables.
>
> > *Sub-Claim:* If $\mathcal{A} \models FO(D)$, then $\mathcal{A} \models FO(D\sigma)$, where D is a clause and σ is a substitution.
> >
> > *Proof*: Easy. For example take $D = \{P(f(x), y), Q(y)\}$, and σ is $(h(x)/y)$. Then $D\sigma = \{P(f(x), h(x)), Q(h(x))\}$. Now $FO(D) = \forall x \forall y (P(f(x), y) \lor Q(y))$, and this logically implies (almost by UI) $FO(D\sigma) = (P(f(x), h(x)) \lor Q(h(x)))$. $\square$
>
> Let σ be the mgu of $C_j^{'} \cup \{B : \neg B \in C_k^{'}\}$. By the sub-claim, $\mathcal{A} \models FO(C_j\sigma)$, and $\mathcal{A} \models FO(C_k\sigma)$. Now $FO(C_j\sigma)$ is equivalent to $\forall x_1 \ldots \forall x_r (B_1 \lor B_2 \lor \ldots \lor B_n)$, and $FO(C_k\sigma)$ is equivalent to $\forall y_1 \ldots \forall y_s (\neg B_1 \lor B_2^{'} \lor \ldots \lor B_n^{'})$. We want to show that $\mathcal{A} \models \forall x_1 \ldots \forall x_r (B_2 \lor \ldots \lor B_n \lor B_2^{'} \lor \ldots \lor B_n{'})$. To prove this, consider all object assignments to $\vec{x}$. For each object assignment, do propositional resolution on $FO(C_j\sigma) \lor FO(C_k\sigma)$. $FO(C_j\sigma)$ is true and $FO(C_k\sigma)$ is true independently of the assignment to B_1.

Completeness part of proof next time.

Prolog

Prolog is based on Horn sentences of the form $\forall \vec{x}(B_1 \land \ldots \land B_k \to A)$ where $B_1, \ldots, B_k$, and A are atomic formulas (no negation). In Prolog, such a formula is written as $A \leftarrow B_1, \ldots, B_k$.

Example: Let the non-logical symbols of the language consist of Knows, a 2-place function, and Father and Mother, two 1-place functions. Consider the following four formulas:

$$\begin{array}{rcl}
\text{Knows}(x, z) & \leftarrow & \text{Knows}(x, y),\ \text{Knows}(y, z) \\
\text{Knows}(x, y) & \leftarrow & \text{Knows}(y, x) \\
\text{Knows}(x, \text{Mother}(x)) & \leftarrow & \\
\text{Knows}(x, \text{Father}(x)) & \leftarrow &
\end{array}$$

The first two formulas are rules for generating new facts from known facts, and the second two formulas consist of the database of known facts.
Now consider the following query:

$$\leftarrow\ \ \text{Knows}(\text{Mother}(x), \text{Father}(x))$$

which expresses "it is not the case that the mother of x knows the father of x." Prolog searches for a contradiction of these five formulas. Finding one yields a proof that the first first four formulas logically imply

$$\forall x(\text{Knows}(\text{Mother}(x), \text{Father}(x))).$$

Prolog, Resolution Theorem

Math 260B - Mathematical Logic

February 13, 1989

Example: (from last lecture)

$$\text{database: } \begin{cases} 1. & \text{Knows}(x,y) \leftarrow \text{Knows}(x,z), \text{Knows}(z,y) \\ 2. & \text{Knows}(x,y) \leftarrow \text{Knows}(y,x) \\ 3. & \text{Knows}(x, \text{Mother}(x)) \leftarrow \\ 4. & \text{Knows}(x, \text{Father}(x)) \leftarrow \end{cases}$$

and a query into the database is:

$$\text{query: } \leftarrow \text{Knows}(\text{Father}(x), \text{Mother}(x))$$

In terms of first order logic, we have

$$\begin{aligned} \Gamma \;=\; \{\; & \forall x \forall y \forall z (\text{Knows}(x,z) \wedge \text{Knows}(z,y) \rightarrow \text{Knows}(x,y)), \\ & \forall x \forall y (\text{Knows}(y,x) \rightarrow \text{Knows}(x,y)), \\ & \forall x (\text{Knows}(x, \text{Mother}(x)), \\ & \forall x (\text{Knows}(x, \text{Father}(x)) \;\} \end{aligned}$$

which capture the meaning of the database, and

$$\begin{aligned} A &= \exists x (\text{Knows}(\text{Father}(x), \text{Mother}(x)) \\ B &= \forall x (\neg \text{Knows}(\text{Father}(x), \text{Mother}(x)) \end{aligned}$$

which captures the "meaning of the query" (quoted since the query really wants to know more than the existence of someone who knows both his or her father and mother, it wants an actual instance of such a person).

Now we have $\Gamma \models A$ iff $\Gamma \cup \{B\}$ is inconsistent. Prolog searches for a refutation as follows:

$$\begin{array}{rll}
1. & \leftarrow & \text{Knows}(\text{Father}(x), z), \text{Knows}(z, \text{Mother}(x)) \\
2. & \leftarrow & \text{Knows}(z, \text{Father}(x)), \text{Knows}(z, \text{Mother}(x)) \\
3. & \leftarrow & \text{Knows}(x, \text{Mother}(x)) \\
4. & \leftarrow &
\end{array}$$

Each refutation step corresponds to a Horn clause. Line 1 resolves the query with the first clause in the database; i.e., it resolves the two Horn clauses

$$\begin{array}{l}
\{\neg\text{Knows}(\text{Father}(x), \text{Mother}(x))\} \\
\{\text{Knows}(x, y), \neg\text{Knows}(x, z), \neg\text{Knows}(z, y)\}
\end{array}$$

to get the Horn clause

$$\{\neg\text{Knows}(\text{Father}(x), z), \neg\text{Knows}(z, \text{Mother}(x))\}.$$

Line 2 resolves line 1 with the second clause in the database. Line 3 resolves line 2 with the fourth clause in the database. Line 4 resolves line 3 with the third clause in the database and derives a contradiction.

The "hoped for" Prolog output in this case is:

$$\text{yes, } x = x.$$

If the fourth clause in the database is changed to

$$\text{Knows}(\text{Ralph}, \text{Father}(\text{Ralph})) \leftarrow,$$

then the "hoped for" output would be:

$$\text{yes, } x = \text{Ralph}.$$

And if a fifth clause was added to the database:

$$\text{Knows}(\text{Abe}, \text{Father}(\text{Abe})) \leftarrow,$$

then the "hoped for" output would be:

$$\begin{array}{l}
\text{yes, } x = \text{Ralph}. \\
\text{yes, } x = \text{Abe}.
\end{array}$$

What's Going On

1. Prolog uses an input (and linear) resolution:
   - The query is the first line in the refutation.
   - The $i+1^{\text{st}}$ clause in the refutation is obtained by resolving the i^{th} clause with a clause from the database.
   - The i^{th} clause is of the form $\leftarrow B_1, \ldots, B_t$ (no un-negated literals).
   - The database clause is of the form $C \leftarrow D_1, \ldots, D_s$.
   - Prolog attempts to unify C and B_1 with an mgu σ and deduce by resolution $\leftarrow D_1\sigma, \ldots, D_s\sigma, B_2\sigma, \ldots, B_2\sigma$.
2. Completeness: Such a refutation exists if the first order translations of the database and the query are inconsistent. (We'll skip the proof since we're going to prove it for Robinson resolution.)
3. Prolog uses a depth-first search for refutations. This doesn't always find a refutation if it exists, but it works well in practice. A breadth-first search would always find one if it exists.
4. By Herbrand's theorem, since $\Gamma \vdash A$ ($A = \exists x A^M(x)$) there are terms $t_1, \ldots, t_k$ such that $\Gamma \vdash A^M(t_1) \vee \ldots \vee A^M(t_k)$. Because Prolog only uses Horn clauses, $k = 1$ suffices. (Since we only use the query once, we only introduce one term, t.) So for $\Gamma \vdash A^M(t)$, the "hoped for" output would be

$$\text{yes, } x = t.$$

Resolution Theorem

Γ is a set of universal sentences with matrices that are disjunctions of literals. There is a Robinson resolution refutation of the set S of clauses expressing Γ iff Γ is inconsistent. Last lecture we proved soundness ($\Rightarrow$); today, completeness.

Proof: ($\Leftarrow$ (completeness)) It suffices to assume that Γ is finite by the compactness theorem. So let $S = \{C_1, \ldots, C_t\}$ where each C_i is a clause. By Gödel's completeness theorem and by Herbrand's theorem there are ground instances $D_1, \ldots, D_t$ that are inconsistent.[1] By the completeness of propositional resolution there is a sequence $D_1, \ldots, D_t, \ldots, D_s$ which is a resolution refutation (i.e., $D_s = \emptyset$) since $D_1, \ldots, D_t$ is inconsistent. What we want to do now is to "lift" this sequence to a Robinson resolution refutation $C_1, \ldots, C_s = \emptyset$.

> *Claim:* Such a Robinson resolution exists with $C_i\sigma_i = D_i$ for some substitution σ_i for $i = 1, \ldots, s$.
>
> *Proof*: By induction on i, $C_1, \ldots, C_i$ can be constructed.
>
> *Basis*: For $i = 1, \ldots, t$ this is clear since we already know such C_i's exist.
>
> *Induction*: Suppose D_i is obtained by resolution from D_j and D_k where $j, k < i$. So $D_i = (D_j \setminus \{A\}) \cup (D_k \setminus \{\neg A\})$ for some literal A. By the induction hypothesis, $D_j = C_j\sigma_j$, and $D_k = C_k\sigma_k$ for some substitutions σ_j and σ_k. (Without loss of generality, for notational convenience, assume that C_j and C_k have no variables in common; otherwise rename to C_k^* and σ_k^*.) Let $C_j^{'} \subseteq C_j$ and $C_k^{'} \subseteq C_k$ be such that
>
> $$\begin{aligned} C_j^{'} &= \{B \in C_j : B\sigma_j = A\}, \text{ and} \\ C_k^{'} &= \{B \in C_k : B\sigma_k = \neg A\}. \end{aligned}$$
>
> Let σ be an mgu of $C_j^{'} \cup \{B : \neg B \in C_k^{'}\}$. σ has as domain a subset of variables in $C_j^{'}$ and $C_k^{'}$ and has a range involving only variables in $C_j^{'}$ and $C_k^{'}$.
>
> (Proof continued next lecture.)

[1] D_i is a ground instance if $D_i = C_i\sigma_i$ for some substitution σ_i and if D_i is variable free.

Resolution Theorem, Craig Interpolation

Math 260B - Mathematical Logic

February 15, 1989

Last time we started the completeness part of the proof of the resolution theorem. So far, we have

- D_i is a resolvent of D_j and D_k with respect to $A \in D_j$ and $\neg A \in D_k$.
- $C_j\sigma_j = D_j$, $C_k\sigma_k = D_k$, and C_j and C_k have no variables in common.
- $C_j^{'} = \{B \in C_j : B\sigma_j = A\}$, and $C_k^{'} = \{B \in C_k : B\sigma_k = \neg A\}$.
- σ is an mgu of $C_j^{'} \cup \{B : \neg B \in C_k^{'}\}$. The domain of σ is a subset of the variables in $C_j^{'}$ and $C_k^{'}$, and terms in the range of σ use only variables in $C_j^{'}$ and $C_k^{'}$.

To continue with the proof, let $\sigma^{'} = \sigma_j \cup \sigma_k$. This makes sense because σ_j and σ_k have disjoint domains. Clearly, $\sigma^{'}$ unifies $C_j^{'} \cup \{B : \neg B \in C_k^{'}\}$. Since σ is an mgu, there is a τ such that $\sigma\tau = \sigma^{'}$. The proof of the following claim will complete the proof of the theorem:

Claim: Let C_i be $(C_j\sigma \setminus C_j^{'}\sigma) \cup (C_k\sigma \setminus C_k^{'}\sigma)$. Then $C_i\tau = D_i$.

Proof:

$$\begin{aligned}
(C_j\sigma \setminus C_j^{'}\sigma)\tau &= \{B\sigma\tau : B\sigma \neq \{A\}\} \\
&= \{B\sigma_j : B\sigma_j \neq \{A\}\} \\
&= D_j \setminus \{A\} \text{ since } D_j = \{B\sigma_j : B \in C_j\}
\end{aligned}$$

Similarly,

$$(C_k\sigma \setminus C_k^{'}\sigma)\tau = D_k \setminus \{\neg A\}.$$

So,

$$\begin{aligned} C_i &= (D_j \setminus \{A\}) \cup (D_k \setminus \{\neg A\}) \\ &= D_i. \ \square \end{aligned}$$

To summarize the process of proving a first order theorem, we first Skolemize to get universal formulas, then we convert the matrices to disjunctive normal form, and finally we apply Robinson resolution.

Some things to note:

- A Robinson resolution refutation takes no more steps than a propositional resolution refutation of the ground instances (the D's).
- The advantage of Robinson resolution is that the terms for the ground instances don't have to be chosen ahead of time.
- The problem of deciding which clauses to resolve exists.
- This isn't a problem with Horn clauses (just use linear resolution), but Horn clauses can't express everything.
- A Horn resolution theorem can be proven using the same technique; i.e., lifting from the propositional case.

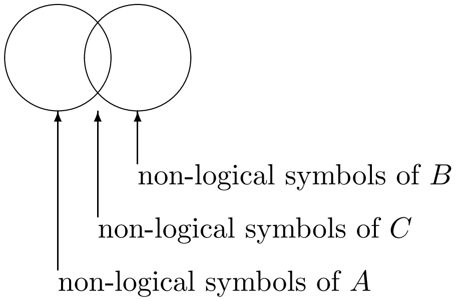
Craig Interpolation

(Due to Bill Craig in the 1940's.)

Craig Interpolation Theorem: Let A and B be first order sentences, and suppose that $A \models B$. Then there is a sentence C such that $A \models C$ and $C \models B$, and such that every non-logical symbol in C appears in both A and B.

Definition: The C in the theorem is called an *interpolant* for A and B.

This theorem is interesting because it says that C distills out the essential non-logical symbols of A which area needed to logically imply B. In a picture, we have


non-logical symbols of B
non-logical symbols of C
non-logical symbols of A

Example: Let A be $\exists x(P(x) \vee \neg P(x))$, and let B be $\exists x(Q(x) \vee \neg Q(x))$. Then certainly $A \models B$ since both A and B are valid. So an interpolant has to be valid; i.e., C could be $\exists x(x = x)$ or $\forall x(x = x)$ or any valid formula using no non-logical symbols.

Example: Let A be $\exists x(P(x) \wedge \neg P(x))$, and let B be $\exists x(Q(x) \wedge \neg Q(x))$. Then $A \models B$ since B is unsatisfiable. Since A is also unsatisfiable, any interpolant has to be unsatisfiable; i.e., C could be $\forall x(x \neq x)$.

Example: A graph G is 2-colorable implies that G doesn't contain a complete subgraph H with 3 vertexes. Let the non-logical symbols of G be E a binary edge relation, R indicating whether or not a vertex is colored red, and B indicating whether or not a vertex is colored blue. Let

$$\begin{aligned} A \quad = \quad & \forall x(R(x) \leftrightarrow \neg B(x)) \ \wedge \\ & \forall x \forall y(xEy \rightarrow (R(x) \leftrightarrow B(y)) \ \wedge \\ & \text{“}E \text{ is irreflexive and symmetric”} \end{aligned}$$

The first part says that every vertex is either red or blue, and the second part says that G is 2-colorable. Let

$$\begin{aligned} B = \neg \exists x_1 \exists x_2 \exists x_3 [\quad & \textstyle\bigwedge_{i=1}^{3} H(x_i) \wedge \\ & \forall y(H(y) \leftrightarrow \textstyle\bigvee_{i=1}^{3} y = x_i) \wedge \\ & x_1 E x_2 \wedge x_2 E x_3 \wedge x_3 E x_1 \ ] \end{aligned}$$

where $H(x)$ is a predicate asserting that x is in subgraph H. An interpolant C will use E as the only non-logical symbol; i.e.

$$C = \neg \exists x_1 \exists x_2 \exists x_3 [x_1 E x_2 \wedge x_2 E x_3 \wedge x_3 E x_1].$$

Craig Interpolation Theorem

Math 260B - Mathematical Logic

February 17, 1989

Craig Interpolation Theorem for Propositional Logic: Suppose that ϕ and ψ are propositional formulas and that $\phi \models \psi$. Then one of the following holds:

1. $\models \psi$.
2. $\models \neg\phi$.
3. There is a ρ such that $\phi \models \rho$ and $\rho \models \psi$ and every propositional variable in ρ also appears in both ϕ and ψ.

(Note that we need cases 1 and 2 since ϕ and ψ may not have any variables in common. If we used the symbols $\top$ and $\bot$ for truth and falsity, then we wouldn't need cases 1 and 2.)

Proof: Suppose that neither case 1 nor 2 holds. So there are truth assignments τ_ψ and τ_ϕ such that $\bar{\tau}_\psi(\psi) = \text{false}$ and $\bar{\tau}_\phi(\phi) = \text{true}$.

> *Claim:* ϕ and ψ have at least one variable in common.
>
> *Proof*: Suppose not. Then let
>
> $$\tau(P) = \begin{cases} \tau_\psi(P) & \text{if } P \text{ appears in } \psi \\ \tau_\phi(P) & \text{if } P \text{ appears in } \phi \\ \text{arbitrary} & \text{otherwise} \end{cases}$$
>
> So $\bar{\tau}(\neg\psi \wedge \phi) = \text{true}$ which is impossible since $\phi \models \psi$. □

Let P be a variable occurring in both ϕ and ψ, let $\top$ be an abbreviation for $(P \vee \neg P)$, and let $\bot$ be an abbreviation for $(P \wedge \neg P)$.

Remark: If every variable in ϕ also appears in ψ, then take $\rho = \phi$.

Proceed by induction on the number of variables that appear in ϕ but not in ψ. Let Q be such a variable, let ϕ_1 be $\phi(\top/Q)$, and let ϕ_2 be $\phi(\bot/Q)$. Note that $Q \models \phi \leftrightarrow \phi_1$ and $\neg Q \models \phi \leftrightarrow \phi_2$. Hence $\phi \wedge (Q \vee \neg Q) \models \phi_1 \vee \phi_2$, and so $\phi \models \phi_1 \vee \phi_2$.

Claim: $\phi_1 \vee \phi_2 \models \psi$.

Proof: Let $\bar{\sigma}(\phi_1 \vee \phi_2) =$ true, let $\sigma^{'}(R) = \sigma(R)$ for any $R \neq Q$, and let

$$\sigma^{'}(Q) = \begin{cases} \text{true} & \text{if } \bar{\sigma}(\phi_1) = \text{ true} \\ \text{false} & \text{otherwise (i.e., } \bar{\sigma}(\phi_2) = \text{ true)} \end{cases}$$

Now, $\bar{\sigma}^{'}(\phi_1 \vee \phi_2) =$ true. Consider two possibilities:

1. $\sigma^{'}(Q) =$ true, $\bar{\sigma}^{'}(\phi_1) =$ true, and hence $\bar{\sigma}^{'}(\phi) =$ true.
2. $\sigma^{'}(Q) =$ true, $\bar{\sigma}^{'}(\phi_2) =$ true, and hence $\bar{\sigma}^{'}(\phi) =$ true.

So $\bar{\sigma}^{'}(\phi) =$ true, hence $\bar{\sigma}^{'}(\psi) =$ true. Now $\bar{\sigma}(\psi) =$ true since σ and $\sigma^{'}$ agree on all variables occurring in ψ. □

So $\phi \models \phi_1 \vee \phi_2$, $\phi_1 \vee \phi_2 \models \psi$, and the number of variables in $\phi_1 \vee \phi_2$ but not in ψ is one less than the number of variables in ϕ but not in ψ. By induction, there is an interpolant ρ for $\phi_1 \vee \phi_2$ and ψ, and $\phi \models \rho$ and $\rho \models \psi$. □

First Order Craig Interpolation Theorem: Suppose that $A \models C$. Then there is a B such that $A \models B$ and $B \models C$ and every non-logical symbol in B appears in both A and C.

Proof: $\mathcal{L}_A$ is the language of A. $\mathcal{L}_C$ is the language of C. If D is of the form $Q_1 x_1 \ldots Q_k x_k D^M$ with D^M quantifier free, then $\bar{D} = \bar{Q}_1 x_1 \ldots \bar{Q}_k x_k \neg D^M$, where $\bar{\forall} = \exists$ and $\bar{\exists} = \forall$. Without loss of generality, assume that A and C are in prenex form. There is a refutation R_0 of $\{A, \bar{C}\}$. Let R_0 be $E_1, \ldots, E_t$. Call E_i an A-formula iff E_i is

1. A,

2. an $\mathcal{L}_A$-equality axiom, or

3. derived by a series of EI and/or UI inferences from such a formula.

Similarly, call E_i a C-formula iff it is not an A-formula; i.e., E_i is

1. $\bar{C}$,
2. an $\mathcal{L}_C$-equality axiom which is not an $\mathcal{L}_A$-equality axiom, or
3. derived by a series of EI and/or UI inferences from such a formula.

Since R_0 is a refutation, $Q_A \cup Q_C$ is tautologically inconsistent where $Q_A = \{\text{quantifier-free } A\text{-formulas}\}$, and $Q_C = \{\text{quantifier-free } C\text{-formulas}\}$. So by Craig interpolation for propositional logic, there is a quantifier-free sentence B_t such that $\{Q_A \cup \neg B_t\}$ and $\{Q_C \cup B_t\}$ are tautologically inconsistent and such that B_t is a propositional combination of atomic sentences that appear in both Q_A and Q_C (assuming that Q_A and Q_C are both tautologically consistent separately). If Q_A is tautologically inconsistent, set $B_t = \forall x(x \neq x)$. If Q_C is tautologically inconsistent, set $B_t = \forall x(x = x)$. Otherwise, $\bigwedge Q_A \models \bigvee Q_C$, and B_t is an interpolant of this.

> *Fact:* There is a refutation of $\neg B_t \cup \{A\text{-formulas in } R_0\}$, and there is a refutation of $B_t \cup \{C\text{-formulas in } R_0\}$.

Every relation symbol in B_t occurs in both $\mathcal{L}_A$ and $\mathcal{L}_C$. Every closed term in B_t occurs in some A-formula in R_0 and in some C-formula in R_0.

(Proof continued next time.)

Craig Interpolation Theorem (cont.)

Math 260B - Mathematical Logic

February 22, 1989

Last time we started the proof of the Craig interpolation theorem. We had $A \models C$, and $R_0 = E_1, \ldots, E_t$ was a refutation of $\{A, \bar{C}\}$. We defined A- and C-formulas, and Q_A and Q_C, the quantifier-free A- and C-formulas. We had $\bigwedge Q_A \models \bigvee \neg Q_C$, and we identified an interpolant B_t such that $\bigwedge Q_A \models B_t$ and $B_t \models \bigvee \neg Q_C$. To continue the proof, we'll start out with some definitions.

Definition: An A-term is a term of the form $f(\ldots)$ where $f \in \mathcal{L}_A \setminus \mathcal{L}_C$ or of the form c where $c \in \mathcal{L}_A \setminus \mathcal{L}_C$ Similarly for a C-term replacing $\mathcal{L}_A \setminus \mathcal{L}_C$ by $\mathcal{L}_C \setminus \mathcal{L}_A$. (Note that for a term of the form $f(\ldots)$, this definition doesn't say anything about the symbols represented by the dots.)

Definition: An occurrence of an A- or C-term is *critical* iff it is not occurring as a proper sub-term of an A- or C-term. (Note that a single A- or C-term can appear critically in one place and not critically in another.) Critical means "maximal" in some sense.

Definition: If E_i is in R_0, then

$$E_i^A \text{ is } \begin{cases} E_i & \text{if } E_i \text{ is an } A\text{-formula} \\ A & \text{otherwise} \end{cases}$$

and

$$E_i^C \text{ is } \begin{cases} E_i & \text{if } E_i \text{ is a } C\text{-formula} \\ \bar{C} & \text{otherwise} \end{cases}$$

Now the proof continues. Without loss of generality, let $E_1 = A$ and let $E_2 = \bar{C}$. So $E_1^A = A$ and $E_1^C = \bar{C}$.

Goal: Construct $B_t, B_{t-1}, \ldots, B_1$ such that the following holds for all $i \leq t$:

1. $\{E_1^A, \ldots, E_i^A, \bar{B}_i\}$ has a refutation.
2. $\{E_1^C, \ldots, E_i^C, B_i\}$ has a refutation.
3. Every critical C-term occurring in B_i also occurs critically in some $E_1^A, \ldots, E_i^A$.
4. Every critical A-term occurring in B_i also occurs critically in some $E_1^C, \ldots, E_i^C$.
5. B_i is of the form $Q_1 z_1 \ldots Q_s z_s B_t^*$ where B_t^* is obtained by changing terms in B_t.

   Claim: The goal will suffice to prove the theorem, since B_1 will be the desired interpolant.

   Proof: We need to show that B_1 satisfies the properties of being an interpolant.

   > *Sub-Claim (i):* Every constant and function symbol in B_1 appears both in A and C.
   >
   > *Proof*: If not, then B_1 would have a critical A- or C-term. This can't happen since $E_1^A = A$ has no C-terms, and $E_1^C = \bar{C}$ has no A-terms. □
   >
   > *Sub-Claim (ii):* $A \models B_1$ and $B_1 \models C$.
   >
   > *Proof*: Both $\{A, \bar{B}_1\}$ and $\{\bar{C}, B_1\}$ have refutations. □
   >
   > *Sub-Claim (iii):* Every relation symbol in B_1 occurs in both A and C.
   >
   > *Proof*: Every relation symbol in B_1 occurs in B_t (by the way B_t was constructed). Hence every such relation symbol occurs in both Q_A and Q_C, and hence in both A and C. □

   So achieving the goal will suffice. □

The following cases show how to build B_i from B_{i+1} and show that the resulting B_i satisfies the properties of the goal above.

Case 0. E_{i+1} is an equality axiom. Then let $B_i \overset{\text{def}}{=} B_{i+1}$. Goal properties 1, 2, and 5 still hold. To show that goal property 3 holds, consider E_{i+1}. If it is not an A-formula, then property 3 holds trivially by the induction hypothesis. Otherwise, E_{i+1} contains no C-terms. So any (critical or not) C-term in $\{E_1^A, \dots, E_{i+1}^A\}$ also occurs in $\{E_1^A, \dots, E_i^A\}$. Goal property 4 holds by a similar argument.

Case 1. E_{i+1} is inferred by EI from the A-formula E_j with $j \leq i$. Then $E_j = \exists x \phi(x)$, $E_{i+1} = \phi(a)$ for some new constant a, and E_{i+1} is also an A-formula. Then let $B_i \overset{\text{def}}{=} \exists x B_{i+1}(x/a)$.

> *Goal 1:* By the induction hypothesis, there is a refutation R of $\{E_1^A, \dots, E_{i+1}^A, \bar{B}_{i+1}\}$. Now, $\bar{B}_i = \forall x \bar{B}_{i+1}(x/a)$. So a refutation of $\{E_1^A, \dots, E_{i+1}^A, \bar{B}_i\}$ contains
>
> $E_j = \exists x \phi(x)$
> then $E_{i+1} = \phi(a)$ by EI
> then $\bar{B}_i = \forall x \bar{B}_{i+1}(x)$ by assumption
> then $\bar{B}_{i+1}$ by UI
> and then R minus any duplicate sentences.
>
> Using EI to get a from x is okay since E_{i+1} appeared in R_0 and was okay. Another satisfactory refutation would be B_i followed by R.

(Proof continued next time.)

Craig Interpolation Theorem (cont.)

Math 260B - Mathematical Logic

February 24, 1989

Last time, we identified the following set of goals for constructing B_t, B_{t-1}, …, B_1 such that for all $i \leq t$:

1. $\{E_1^A, \ldots, E_i^A, \bar{B}_i\}$ has a refutation.
2. $\{E_1^C, \ldots, E_i^C, B_i\}$ has a refutation.
3. Every critical C-term occurring in B_i also occurs critically in some $E_1^A, \ldots, E_i^A$.
4. Every critical A-term occurring in B_i also occurs critically in some $E_1^C, \ldots, E_i^C$.
5. B_i is of the form $Q_1 z_1 \ldots Q_s z_s B_t^*$ where B_t^* is obtained by changing terms in B_t.

We also proved the first and started the proof of the second of five cases to show how to build B_i from B_{i+1}:

Case 1. E_{i+1} is $\phi(a)$, E_j is $\exists x \phi(x)$, E_{i+1} is inferred from E_j by EI, and both E_{i+1} and E_j are A-formulas. Then $B_i \overset{\text{def}}{=} \exists x B_{i+1}(x/a)$.

Goal 1: Satisfied last time.
Goal 2: By the induction hypothesis, $\{E_1^C, \ldots, E_{i+1}^C, B_{i+1}\}$ has a refutation R. A refutation of $\{E_1^C, \ldots, E_i^C, B_i\}$ is

B_i
B_{i+1} by EI
R (minus B_{i+1})

Note that there is no problem with E^C_{i+1} not appearing in R, since E_{i+1} is an A-term, and so $E^C_{i+1} = \bar{C} = E^C_1$.

Goal 3:

> *Claim:* A critical C-term in B_i cannot contain x. Similarly, a critical C-term in B_{i+1} cannot contain a.
>
> *Proof*: Suppose that $t(x)$ is a critical C-term in B_i. Then $t(a)$ is a critical C-term in B_{i+1}. So $t(a)$ appears (critically) in $E^A_1, \ldots, E^A_{i+1}$ by the induction hypothesis. Since a cannot appear in $E_1, \ldots, E_i$, $t(a)$ occurs in E_{i+1}, and hence $t(x)$ appears in E_j. Now E_j was ultimately derived from A (in R_0) by EI and UI inferences. Let $t(x)$ be $f_C(\ldots x \ldots)$, where $f_C \in \mathcal{L}_C \setminus \mathcal{L}_A$. So $f_C(\text{—}x\text{—})$ must appear in A, where "—x—" is "$\ldots x \ldots$" with some closed sub-terms changed to variables by going backwards through EI and/or UI inferences. But this is a contradiction since $f_C \notin \mathcal{L}_A$. □

By the claim, every critical C-term in B_i occurs critically in B_{i+1} and in $E^A_1, \ldots, E^A_{i+1}$. Since they occur in E^A_{i+1}, then they also occur in E^A_j and thus in $E^A_1, \ldots, E^A_i$.

Goal 4:

> *Claim:* Any critical A-term in B_i also occurs critically in B_{i+1}.
>
> *Proof*: Suppose that $t(x)$ is a critical A-term in B_i. Then $t(a)$ is a critical A-term in B_{i+1}. So $t(a)$ must appear (critically) by the induction hypothesis in $\{E^C_1, \ldots, E^C_i\}$. But this is impossible since a was introduced by an EI inference and cannot appear in $E_1, \ldots, E_i$. □

This suffices to satisfy goal 4 since $\{E^C_1, \ldots, E^C_{i+1}\} = \{E^C_1, \ldots, E^C_i\}$.

Goal 5: is trivial.

Case 2. E_{i+1} is $\phi(a)$, E_j is $\exists x\phi(x)$, E_{i+1} is inferred from E_j by EI, and both E_{i+1} and E_j are C-formulas. Then $B_i \overset{\text{def}}{=} \forall x B_{i+1}(x/a)$.

In this case, the roles of B_i and $\bar{B}_i$ are swapped, A- and C-formulas are swapped, and the proofs of goals 1, 2, 3, and 4 are the duals of the proofs of goals 2, 1, 4, and 3 respectively from case 1.

Case 3. E_{i+1} is $\phi(t)$, E_j is $\forall x\phi(x)$, E_{i+1} is inferred from E_j by UI, and both E_{i+1} and E_j are A-formulas. Let $r_1, \ldots, r_k$ be the C-terms that occur critically in B_{i+1} and E^A_{i+1}, but not critically in $E^A_1, \ldots, E^A_i$. Then $B_i \overset{\text{def}}{=} \forall y_1 \ldots \forall y_k B_{i+1}(y_1/r_1, \ldots, y_k/r_k)$. (Note the abuse of notation y_i/r_i. This really means substitute y_i for critical occurrences of the terms r_i. We will also abbreviate a series of such substitutions by, for example, $B_{i+1}(\vec{y}/\vec{r})$.)

Goal 1: By the induction hypothesis, there is a refutation R of $\{E^A_1, \ldots, E^A_{i+1}, \bar{B}_{i+1}\}$. Now $\bar{B}_i$ is $\exists y_1 \ldots \exists y_k \bar{B}_{i+1}(\vec{y}/\vec{r})$. So let R^* be the refutation:

$\bar{B}_i$
$\exists y_2 \ldots \exists y_k \bar{B}_{i+1}(a_1/r_1)$ by EI
$\vdots$
$\bar{B}_{i+1}(a_1/r_1, \ldots, a_k/r_k)$ by EI
$R(a_1/r_1, \ldots, a_k/r_k)$ minus the duplicate $\bar{B}_{i+1}(\vec{a}/\vec{r})$'s.

with the added condition that E^A_j appear before E^A_{i+1}.

Claim: R^* is a refutation of $\{E^A_1, \ldots, E^A_i, \bar{B}_i\}$.

Proof: Note that none of $E^A_1, \ldots, E^A_i$ get modified by the above above transformation since, by definition, they don't contain any of the r_i's. Also, note that no r_i is a constant introduced by EI since the r_i's are C-terms. So the uses of EI are okay in R^*. The only way a use of UI could cause problems in R^* is if $\forall x D(s(x))$ was used to derive $D(s(v))$ in R where $s(v) = r_i$. But this is impossible since we can't get rid of a C-term by going backwards in R, and $s(x)$ would have a C-function symbol as its outermost symbol. Finally, note that changing critical r_i's to a_i's is a 1-1 transformation of atomic sentences. So the quantifier-free sentences of R^* are still tautologically inconsistent. □

Craig Interpolation Theorem (cont.), Beth Definability

Math 260B - Mathematical Logic

February 27, 1989

Last time we were in the middle of proving case 3.

Case 3. E_{i+1} is $\phi(t)$, E_j is $\forall x\phi(x)$, E_{i+1} is inferred from E_j by UI, and both E_{i+1} and E_j are A-formulas. $r_1, \ldots, r_k$ are the C-terms that occur critically in B_{i+1} and E^A_{i+1}, but not critically in $E^A_1, \ldots, E^A_i$. $B_i \overset{\text{def}}{=} \forall y_1 \ldots \forall y_k B_{i+1}(\vec{y}/\vec{r})$.

Goal 1: That $\{E^A_1, \ldots, E^A_i, \bar{B}_i\}$ has a refutation was shown last time.

Goal 2: By the induction hypothesis, $\{E^C_1, \ldots, E^C_{i+1}, B_{i+1}\}$ has a refutation R. A refutation of $\{E^C_1, \ldots, E^C_i, B_i\}$ is:

$$\begin{array}{ll} \forall y_1 \ldots \forall y_k B_{i+1}(\vec{y}/\vec{r}) & (= B_i) \\ \forall y_2 \ldots \forall y_k B_{i+1}(\vec{y}/\vec{r}) & \text{by UI} \\ \vdots & \\ B_{i+1} \text{ by UI} & \\ R & \end{array}$$

i.e., we first use k UI inferences to change the y_i's back to the old r_i's and then continue with R. (Note that $E^C_{i+1} = \bar{C} = E^C_1$.)

Goal 3: Any critical C-term in B_i appears in $\{E^A_1, \ldots, E^A_i\}$.

Claim: No critical C-term in B_i contains any y_j.

Proof: r_j is critical in B_{i+1}, so y_j is not inside a C-term in B_i. □

By the claim, any critical C-term in B_i is critical in B_{i+1} and is not one of the r_j's (if it were, it would have been changed to y_j), and thus appears in $E_1^A, \ldots, E_i^A$ by definition of $r_1, \ldots, r_k$. (Note that we pick the r_i's to satisfy goal 3.)

Goal 4: (similar to goal 3) Any critical A-term in B_i appears in $\{E_1^C, \ldots, E_i^C\}$.

> *Claim again:* No critical A-term in B_i contains any y_j.
>
> *Proof*: (same)

By the claim, any critical C-term in B_i is critical in B_{i+1} This suffices to satisfy the goal since $\{E_1^C$, ..., $E_{i+1}^C\} = \{E_1^C$, ..., $E_i^C\}$.

Goal 5: Obvious.

Case 4. E_{i+1} is $\phi(t)$, E_j is $\exists x \phi(x)$, E_{i+1} is inferred from E_j by EI, and both E_{i+1} and E_j are C-formulas. $r_1, \ldots, r_k$ are the A-terms that occur critically in B_{i+1} and E_{i+1}^C, but not critically in $E_1^C, \ldots, E_i^C$. $B_i \overset{\text{def}}{=} \exists y_1 \ldots \exists y_k B_{i+1}(\vec{y}/\vec{r})$, so $\bar{B}_i = \forall y_1 \ldots \forall y_k \bar{B}_{i+1}(\vec{y}/\vec{r})$. The rest is the dual of case 3.

This concludes the proof of the first order version of Craig's interpolation theorem. □ (Whew!)

Note that the proof gives an explicit method for forming an interpolant. Furthermore, refutations of $A \models B_1$ and $B_1 \models C$ will be a lot like a refutation of $A \models C$.

Beth Definability

Let $\mathcal{L}$ be a language. Let P and $P^{'}$ be two additional k-ary relation symbols not in $\mathcal{L}$. Let $\Gamma(P)$ be a set of sentences in the language $\mathcal{L} \cup \{P\}$. And let $\Gamma(P^{'})$ be a set of sentences $\Gamma(P)$ with all P's changed to $P^{'}$'s.

Definition: $\Gamma(P)$ *explicitly defines* P iff there is a formula $\phi(x_1, \ldots, x_k)$ in the language $\mathcal{L}$ such that $\Gamma(P) \models \forall \vec{x}(P(\vec{x}) \leftrightarrow \phi(\vec{x}))$ for all $\vec{x}$. I.e., P can be explicitly defined in terms of symbols of $\mathcal{L}$.

Definition: $\Gamma(P)$ *implicitly defines* P iff

$$\Gamma(P) \cup \Gamma(P') \models \forall\vec{x}(P(\vec{x}) \leftrightarrow P'(\vec{x})).$$

Equivalently, if $\mathcal{M} \models \Gamma(P)$, $\mathcal{N} \models \Gamma(P')$, $|\mathcal{M}| = |\mathcal{N}|$, and $\mathcal{M}$ and $\mathcal{N}$ agree on the interpretation of symbols of $\mathcal{L}$, then $P^{\mathcal{M}} = P'^{\mathcal{N}}$.

Beth Definability Theorem: $\Gamma(P)$ explicitly defines P iff it implicity defines P.

Proof: $(\Rightarrow)$ Trivial.

$(\Leftarrow)$ By compactness, there is a finite $\Gamma_0 \subseteq \Gamma$ such that

$$\Gamma_0(P) \cup \Gamma_0(P') \models \forall\vec{x}(P(\vec{x}) \leftrightarrow P'(\vec{x})).$$

(Now we want to transform this so we can use Craig interpolation.) Let $c_1, \dots, c_k$ be new constant symbols. In particular, note that

$$\Gamma_0(P) \cup \Gamma_0(P') \models (P(\vec{c}) \leftrightarrow P'(\vec{c})).$$

So

$$(\bigwedge \Gamma_0(P)) \wedge P(\vec{c}) \models (\bigwedge \Gamma_0(P')) \rightarrow P'(\vec{c}).$$

By the Craig interpolation theorem, there is a $B(x_1, \dots, x_k)$ in the language $\mathcal{L}$ such that

$$\begin{aligned} (\bigwedge \Gamma_0(P)) \wedge P(\vec{c}) &\models B(\vec{c}), \text{ and} && (1) \\ B(\vec{c}) &\models (\bigwedge \Gamma_0(P')) \rightarrow P'(\vec{c}). && (2) \end{aligned}$$

From (2), we can rename P' to P without affecting its truth to get

$$B(\vec{c}) \models (\bigwedge \Gamma_0(P)) \rightarrow P(\vec{c}).$$

By this and (1),

$$\begin{aligned} \Gamma_0(P) &\models P(\vec{c}) \rightarrow B(\vec{c}), \text{ and} \\ \Gamma_0(P) &\models B(\vec{c}) \rightarrow P(\vec{c}). \end{aligned}$$

By a previous homework problem,

$$\begin{aligned}\Gamma_0(P) &\models \forall\vec{x}(P(\vec{x}) \to B(\vec{x})), \text{ and}\\ \Gamma_0(P) &\models \forall\vec{x}(B(\vec{x}) \to P(\vec{x})),\end{aligned}$$

and so

$$\Gamma_0(P) \models \forall\vec{x}(B(\vec{x}) \leftrightarrow P(\vec{x})).\ \Box$$

To get an intuitive idea of what this theorem is saying, suppose that we're working in a theory with language L, and we're trying to extend the theory to incorporate a new concept P. If it is possible to add statements (in the language L plus P) about original concepts in L plus the new concept P which implicitly define P (uniquely specify in all models), then P can be explicitly defined.

Introduction to Model Theory

Math 260B - Mathematical Logic

March 1, 1989

Model theory views logic from the point of view of structures instead of proofs.

Definition: A *theory* is a set of sentences. A theory is *closed* iff it is closed under $\models$. (Closure under $\models$ means that if T is a theory and $T \models A$, then $A \in T$.)

In many definitions, a theory *is* a closed theory. We will adopt this convention. We have briefly been introduced to theories in the past. The theory of groups is the set of logical consequences of the axioms for groups. The theory of a structure is the set of sentences true for that structure; i.e., $Th(N, 0, S, +, \cdot)$.

Example: Dense linear order (DLO) without endpoints. Here, the only non-logical symbol will be $\leq$. For notational convenience, let $s < t$ be an abbreviation for $s \leq t \wedge s \neq t$. The axioms are:

1. $\forall x \forall y (x \leq y \vee y \leq x)$
2. $\forall x \forall y \forall z (x \leq y \wedge y \leq z \rightarrow x \leq z)$
3. $\forall x \forall y (x \leq y \wedge y \leq x \rightarrow x = y)$
4. $\forall x \forall y (x < y \rightarrow \exists z (x < z \wedge z < y))$
5. $\forall x \exists y (y < x)$
6. $\forall x \exists y (x < y)$

The first three axioms express linear order, the fourth axiom expresses density, and the last two axioms express no left and right endpoints. Some models of these axioms are:

$$\begin{aligned} ((0,1),\leq) &\models \text{DLO} \quad (1)\\ (Q,\leq) &\models \text{DLO} \quad (2)\\ ((Q\cap(0,1))\cup(4,5),\leq) &\models \text{DLO} \quad (3)\\ ((0,1)\cup(1,2),\leq) &\models \text{DLO} \quad (4) \end{aligned}$$

Furthermore, no two of these structures are isomorphic. (1) is the set of reals between 0 and 1. (2) is the set of rationals whose cardinality is less than that of (1). (3) is the set of rationals between 0 and 1 plus the set of reals between 4 and 5; this is not isomorphic to (1) because any mapping of the rationals between 0 and 1 to a sub-interval of the reals between 0 and 1 will not be onto. (4) is the set of reals between 0 and 1 plus the set of reals between 1 and 2. This is not isomorphic to (1) because any mapping from (4) to (1) will leave out at least one point; i.e. the point that the real number 1 would map to.

Lemma: Any two countable models of DLO without endpoints are isomorphic.

Proof: First note that any countable dense model has to be infinite. Suppose that $\mathcal{A} \models (A, \leq^{\mathcal{A}})$ and $\mathcal{B} \models (B, \leq^{\mathcal{B}})$ are two countable models of DLO without endpoints. Since A and B are countable, we can enumerate them distinctly as

A: $\{a_1, a_2, \ldots\}$ and
B: $\{b_1, b_2, \ldots\}$.

Now we can't just map a_i to b_i because we have to respect $\leq$. So we'll define $f : \mathcal{A} \cong \mathcal{B}$ as follows in stages.

Stage 1. $f(a_1) =$ some arbitrary b, say b_1.

Stage $2i-1$**.** If $f(a_i)$ is already defined, go to stage $2i$. Otherwise, let $a_{k_1}, \ldots, a_{k_j}$ be the a's at which f is already defined and ordered so that

$$a_{k_1} <^{\mathcal{A}} a_{k_2} <^{\mathcal{A}} \cdots <^{\mathcal{A}} a_{k_j}.$$

Now

$$f(a_{k_1}) <^{\mathcal{B}} f(a_{k_2}) <^{\mathcal{B}} \cdots <^{\mathcal{B}} f(a_{k_j})$$

by the way f is constructed. So define $f(a_i) =$ some arbitrary b such that

$$\begin{aligned} f(a_{k_s}) <^{\mathcal{B}} b <^{\mathcal{B}} f(a_{k_{s+1}}) &\quad \text{if} \quad a_{k_s} <^{\mathcal{A}} a_i <^{\mathcal{A}} a_{k_{s+1}}, \\ b <^{\mathcal{B}} f(a_{k_1}) &\quad \text{if} \quad a_i <^{\mathcal{A}} a_{k_1}, \text{ or} \\ f(a_{k_j}) <^{\mathcal{B}} b &\quad \text{if} \quad a_{k_j} <^{\mathcal{A}} a_i. \end{aligned}$$

Stage $2i$. If b_i is already in the range of f go to stage $2i+1$. Otherwise, let

$$f(a_{k_1}) <^{\mathcal{B}} f(a_{k_2}) <^{\mathcal{B}} \cdots <^{\mathcal{B}} f(a_{k_s})$$

be the values already in the range of f. Now pick a arbitrarily such that a is ordered with respect to $a_{k_1}, \ldots, a_{k_s}$ in the same way as b is with respect to $f(a_{k_1}), \ldots, f(a_{k_s})$ and such that $f(a)$ has not yet been defined. Define $f(a) = b$.

The odd stages insure that the domain of f will be all of the a's, and the even stages insure that the range of f will be all of the b's. After going through stages $i = 1, 2, \ldots$ we get a 1-1 onto function $f : A \to B$ such that f preserves $\leq$. $f(a_i)$ is the value assigned to a_i at or before stage $2i$. □

Definition: A theory T is *complete* iff for every sentence A in the language of T, either $T \models A$ or $T \models \neg A$. (An inconsistent theory is complete, but generally our theories are consistent.)

Theorem: DLO without endpoints is a complete theory.

Proof: Let T be DLO without endpoints. Suppose that T is not complete and that A is such that $T \not\models A$ and $T \not\models \neg A$. So $T \cup \{\neg A\}$ and $T \cup \{A\}$ are consistent. By the Löwenheim-Skolem theorem, $T \cup \{\neg A\}$ and $T \cup \{A\}$ have countable models, say $\mathcal{A}$ and $\mathcal{B}$ respectively. (Note that since $\mathcal{A} \models T \cup \{\neg A\}$, $\mathcal{A} \models T$, and similarly, $\mathcal{B} \models T$.) But by the above lemma, $\mathcal{A} \cong \mathcal{B}$, which is impossible since isomorphisms preserve truth of sentences and $\mathcal{A}$ and $\mathcal{B}$ don't satisfy the same set of first order sentences. □

Definition: A theory T is *α-categorical* iff T has exactly one model of cardinality α (up to isomorphism).

Łos-Vaught Test: If T is a theory in a countable language, has no infinite models, and is ω-categorical ($\aleph_0$-categorical), then T is complete.

Note that the proof of the above theorem essentially derives this test.

Łos-Vaught Test, Elementarily Equivalence, Elimination of Quantifiers

Math 260B - Mathematical Logic

March 3, 1989

Last time we introduced the Łos-Vaught Test for countable languages. Today we'll start with the general test.

Definition: T is *α-categorical* iff all models of T of cardinality α are isomorphic.

Definition: If $\mathcal{L}$ is a language, let $||\mathcal{L}|| = \max\{\omega,\ \text{cardinality of } \mathcal{L}\}$.

Łos-Vaught Test: If T is a theory with language $\mathcal{L}$, if $\alpha \geq ||\mathcal{L}||$, if T is α-categorical, and if T has no finite models, then T is complete.

Proof: Suppose not, and suppose that $T \cup \{A\}$ and $T \cup \{\neg A\}$ are both consistent. Then there are structures $\mathcal{A}$ and $\mathcal{B}$ such that $\mathcal{A} \models T \cup \{A\}$ and $\mathcal{B} \models T \cup \{\neg A\}$. Since $\mathcal{A}$ and $\mathcal{B}$ are infinite, by the Löwenheim-Skolem theorem, there are structures $\mathcal{A}^*$ and $\mathcal{B}^*$ of cardinality α such that $\mathcal{A}^* \models T \cup \{A\}$ and $\mathcal{B}^* \models T \cup \{\neg A\}$. Now $\mathcal{A}^* \not\cong \mathcal{B}^*$ which is a contradiction. □

Elementarily Equivalence

Definition: Let $\mathcal{A}$ and $\mathcal{B}$ be structures for a language $\mathcal{L}$. $\mathcal{A}$ and $\mathcal{B}$ are *elementarily equivalent* (written $\mathcal{A} \equiv \mathcal{B}$), iff for every sentence ϕ in the language $\mathcal{L}$, $\mathcal{A} \models \phi$ iff $\mathcal{B} \models \phi$.

Clearly, T is complete iff every pair of its models are elementarily equivalent. Also, in the Łos-Vaught test, "α-categorical" can be weakened to "any two models of T of cardinality α are elementarily equivalent." An example of why this is weaker is DLO without endpoints. There are both countable and uncountable models of DLO without endpoints. These are all elementarily equivalent but not isomorphic. Also, DLO without endpoints is not $2^{\aleph_0}$-categorical by examples from the previous lecture.

Elimination of Quantifiers

Definition: $T \models B(\vec{x})$ iff $T \models \forall\vec{x}B(\vec{x})$.

Definition: A theory T *admits elimination of quantifiers* iff for every formula $A(x_1, \ldots, x_n)$ there is a quantifier-free formula $\phi(x_1, \ldots, x_m)$ such that

$$T \models A(x_1, \ldots, x_n) \leftrightarrow \phi(x_1, \ldots, x_m).$$

Example: If $Th(N, 0, S, +, \cdot)$ admitted elimination of quantifiers (which it doesn't), then there would have been a quantifier-free formula $\phi(x)$ which is equivalent to "x is prime"; i.e.,

$$Th(N, 0, S, +, \cdot) \models \forall x(\text{“}x \text{ is prime”} \leftrightarrow \phi(x)).$$

Proposition: Without loss of generality in the previous definition,

$$\begin{array}{ll} m \leq n & \text{if } n \geq 1, \text{ and} \\ m = 1 & \text{if } n = 0. \end{array}$$

Proof: Suppose that $n \geq 0$, $m > n$, and

$$T \models \forall x_1, \ldots, \forall x_m(A(x_1, \ldots, x_n) \leftrightarrow \phi(x_1, \ldots, x_n, x_{n+1}, \ldots, x_m)).$$

Then

$$T \models \forall x_1, \ldots, \forall x_n(A(x_1, \ldots, x_n) \leftrightarrow \phi(x_1, \ldots, x_n, x_1, \ldots, x_1)). \ \Box$$

The reason that we need the special case for $n = 0$ is that the language may not have constant symbols. For example if A is a sentence and $T \models A$, then one possibility fitting the above the definition is $T \models (A \leftrightarrow x = x)$.

Theorem: To show that T admits elimination of quantifiers, it suffices to show that for every formula of the form $\exists yA$ with A quantifier-free, there is a quantifier-free formula ϕ such that $T \models \exists yA \leftrightarrow \phi$.

An example of the use of this theorem is to go from

$$\begin{array}{rl} \forall x \exists y A & \text{to} \\ \forall x \phi & \text{to} \\ \neg \exists x (\neg \phi) & \text{and then to} \\ \neg \psi & \end{array}$$

where A, ϕ, and ψ are quantifier-free. The idea is to eliminate the innermost quantifiers first and work outward.

Elimination of Quantifiers

Math 260B - Mathematical Logic

March 6, 1989

Some examples of elimination of quantifiers from DLO (from now on, let DLO mean DLO without endpoints):

Example:

$$A(z) = \exists x \forall y (y \leq x \leftrightarrow y \leq z).$$

We want a formula $\phi(z)$ such that DLO $\models \forall z(A(z) \leftrightarrow \phi(z))$. In this example, $\models A(z)$ (since we can always just take $x = z$), and in particular, DLO $\models A(z)$. So DLO $\models A(z) \leftrightarrow z \leq z$.

Example:

$$A(y,z) \; = \; \exists x (y \leq x \wedge x \leq z).$$

DLO $\models A(y,z) \leftrightarrow y \leq z$.

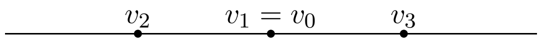
Example:

$$A(v_1, v_2) \; = \; \exists z \forall y ((y \leq z \rightarrow y \leq v_1) \wedge (z \leq y \rightarrow v_2 \leq y)).$$

DLO $\models A(v_1, v_2) \leftrightarrow v_1 \leq v_2$.

Definition: Let $v_0, \ldots, v_n$ be distinct variables. An *arrangement* of $v_0, \ldots, v_n$ is a formula of the form $\theta_0 \wedge \ldots \wedge \theta_{n-1}$ where each θ_i is either $u_i = u_{i+1}$ or $u_i < u_{i+1}$ and $\{u_0, \ldots, u_n\} = \{v_0, \ldots, v_n\}$. (Informally, $u_0, \ldots, u_n$ are the same variables in a possibly different order.)

Example: $v_2 < v_1 \wedge v_1 = v_0 \wedge v_0 < v_3$. Here, u_0 is v_2, u_1 is v_1, u_2 is v_0, and u_3 is v_3. Pictorially then,



Lemma 1: Let A be an arrangement of $v_0, \ldots, v_n$, and let B be of the form $v_i = v_j$ or $v_i \leq v_j$. Then either DLO $\models A \to B$, or DLO $\models A \to \neg B$.

Proof: Obvious.

Lemma 2: If A is an arrangement of $v_0, \ldots, v_n$ and B is a quantifier-free formula with free variables among $v_0, \ldots, v_n$, then either DLO $\models A \to B$, or DLO $\models A \to \neg B$.

Proof: If B is atomic, then lemma 1 applies. So proceed by induction on the complexity of B.

Case 1. B is $\neg C$. Easy.

Case 2. B is $C \wedge D$. Then there are 4 sub-cases to check.

$\vdots$

Definition: If $A(v_0, \ldots, v_n)$ is an arrangement, then B and A are *compatible* iff there is a model $\mathcal{A}$ of DLO with objects $a_0, \ldots, a_n \in |\mathcal{A}|$ such that $\mathcal{A} \models B(a_0, \ldots, a_n) \wedge A(a_0, \ldots, a_n)$.

Lemma 3: Suppose that $B(v_0, \ldots, v_n)$ is a quantifier-free formula (with only the indicated free variables), and suppose that DLO $\not\models \neg B$. Then there is a formula ϕ, which is a disjunction of arrangements of $v_0, \ldots, v_n$, such that DLO $\models B \leftrightarrow \phi$. (Note that this is a special kind of disjunctive normal form; i.e., one that also specifies order.)

Proof:

> *Note 1:* There are finitely many arrangements of $v_0, \ldots, v_n$.
>
> *Note 2:* If $\mathcal{A} \models$ DLO and $a_0, \ldots, a_n \in |\mathcal{A}|$, then there is an arrangement A of $v_0, \ldots, v_n$ such that $\mathcal{A} \models A(a_0, \ldots, a_n)$.
>
> *Claim:* A and B are compatible iff DLO $\models A \to B$.
>
> *Proof*: ($\Rightarrow$) If DLO $\not\models A \to B$, then by lemma 2, DLO $\models A \to \neg B$ which contradicts A and B being compatible.

> ($\Leftarrow$) Every arrangement is satisfiable in a model of DLO. So there is a model $\mathcal{A}$ of DLO with $a_0, \ldots, a_n \in |\mathcal{A}|$ such that $\mathcal{A} \models A(a_0, \ldots, a_n)$. Since DLO $\models A \to B$, $\mathcal{A} \models B(a_0, \ldots, a_n)$. So A and B are compatible. □

(Proof of lemma 3 continued.) Take ϕ to be the disjunction of arrangements compatible with B. By the claim, DLO $\models \phi \to B$. And by the definition of ϕ, DLO $\models B \to \phi$. (I.e., in any model $\mathcal{A}$ of DLO and for any $a_0, \ldots, a_n \in |\mathcal{A}|$, if $\mathcal{A} \models B(a_0, \ldots, a_n)$, then, by Note 2, there is an arrangement A such that $\mathcal{A} \models A(a_0, \ldots, a_n)$; this arrangement is one of the disjuncts in ϕ.) □

Theorem: DLO without endpoints admits elimination of quantifiers.

Proof: By an earler result, it suffices to consider $\exists v_n \phi(v_0, \ldots, v_n)$ where ϕ is quantifier-free and $n \geq 0$. If ϕ is disprovable, then DLO $\models \neg\phi(v_0, \ldots, v_n)$. So

$$\text{DLO} \models \neg\exists v_n \phi(v_0, \ldots, v_n),$$

and so

$$\text{DLO} \models \exists v_n \phi(v_0, \ldots, v_n) \leftrightarrow (v_0 \not\leq v_0).$$

So assume that ϕ is not disprovable. Then by lemma 3,

$$\phi \leftrightarrow \bigvee_i A_i(v_0, \ldots, v_n)$$

where each A_i is an arrangement of $v_0, \ldots, v_n$. And so,

$$\exists v_n \phi(v_0, \ldots, v_n) \leftrightarrow \bigvee_i \exists v_n A_i(v_0, \ldots, v_n).$$

So it suffices to show that $\exists v_n A(v_0, \ldots, v_n)$, where A is an arrangement of $v_0, \ldots, v_n$ is equivalent (provably in DLO) to a quantifier-free formula.

(Proof continued next time.)

Elimination of Quantifiers, Substructures

Math 260B - Mathematical Logic

March 8, 1989

Theorem: DLO without endpoints admits elimination of quantifiers.

Proof: (Continued from last time.) Last time we got to the point where it sufficed to show that if $A(v_0, \ldots, v_{n+1})$ is an arrangement, then

$$\exists v_{n+1} A(v_0, \ldots, v_{n+1})$$

is equivalent (with respect to DLO) to a quantifier-free formula.

Claim: There is a unique arrangement $A^*(v_0, \ldots, v_n)$ such that

$$\begin{aligned} \text{DLO} &\models A \to A^* && (1) \\ \text{DLO} &\models A^*(v_0, \ldots, v_n) \to \exists v_{n+1} A(v_0, \ldots, v_{n+1}) && (2) \end{aligned}$$

Proof: (Informal.) (1) To get A^* from A, we essentially remove v_{n+1} from the ordering specified by A; i.e., if in A we have

$$\ldots\ v_i < v_{n+1} \ \wedge\ v_{n+1} < v_j\ \ldots,$$

then in A^* we get

$$\ldots\ v_i < v_j\ \ldots$$

Considerations have to be made if v_{n+1} appears at either end of the arrangement A. Also, care has to be taken if v_{n+1} is related to variables by '=' instead of '<'.

(2) is true because of the density property of DLO; i.e., we can put v_{n+1} into the order specified by A^* at the same place where we removed it from A to get A^*. □

Hence,

$$\text{DLO} \models \exists v_{n+1} A(v_0, \ldots, v_{n+1}) \leftrightarrow A^*(v_0, \ldots, v_n). \ \square$$

Corollary: DLO without endpoints is complete.

Proof: Given any *sentence* A there is a quantifier-free *formula* $B(v_0)$ such that DLO $\models A \leftrightarrow B(v_0)$. (Note that B only has one free variable; this is a result of the construction from the above theorem.) Now DLO $\models v_0 = v_0$, and DLO $\models v_0 \leq v_0$. And so by induction on the complexity of $B(v_0)$, either DLO $\models B(v_0)$ or DLO $\models \neg B(v_0)$.

Corollary: DLO without endpoints is decidable. (I.e., the set of logical consequences of the axioms is decidable.)

Proof: Given a sentence A, we want to know if DLO $\models A$. By the proof of the above theorem, there is an effective method to get a $B(v_0)$ such that DLO $\models A \leftrightarrow B(v_0)$ and B is quantifier-free. Determining if DLO $\models B(v_0)$ is easy by the proof of the previous corollary. $\square$

Fact: There are theories which admit elimination of quantifiers, but which are not complete.

Substructures

Recall that $\mathcal{A} \equiv \mathcal{B}$ ($\mathcal{A}$ and $\mathcal{B}$ are elementarily equivalent) means that $\mathcal{A}$ and $\mathcal{B}$ are structures for the same language and satisfy the same set of sentences.

Definition: $\mathcal{A}$ is a *substructure* of $\mathcal{B}$, written $\mathcal{A} \subseteq \mathcal{B}$, iff

1. $|\mathcal{A}| \subseteq |\mathcal{B}|$, and
2. the interpretations of non-logical symbols of $\mathcal{A}$ agree with those of $\mathcal{B}$; i.e.,
   (a) $c^{\mathcal{A}} = c^{\mathcal{B}}$ for all constant symbols c,
   (b) $P^{\mathcal{A}} = P^{\mathcal{B}} \cap |\mathcal{A}|^k$ for all k-ary predicates P, and
   (c) $f^{\mathcal{A}} = f^{\mathcal{B}}$ restricted to $|\mathcal{A}|$ for all functions f.

Example: Let $\mathcal{A} = ((0,1), \leq)$, and $\mathcal{B} = ((0,1], \leq)$. Then $\mathcal{A} \subseteq \mathcal{B}$. Note that in $\mathcal{B}$ you can define something that is not in $\mathcal{A}$, but you can't give a closed term for it; i.e. $\exists x \forall y (y \leq x)$.

Example: Subgroups are substructures in the language of groups.

Definition: $\mathcal{A}$ is an *elementary substructure* of $\mathcal{B}$, written $\mathcal{A} \prec \mathcal{B}$, iff $\mathcal{A} \subseteq \mathcal{B}$, and for all elements $a_1, \ldots, a_n \in |\mathcal{A}|$ and for all formulas $\phi(x_1, \ldots, x_n)$, $\mathcal{A} \models \phi(a_1, \ldots, a_n)$ iff $\mathcal{B} \models \phi(a_1, \ldots, a_n)$.

Example: Let $\mathcal{A} = ((0,1), \leq)$, and $\mathcal{B} = ((0,1], \leq)$. Then $\mathcal{A} \not\prec \mathcal{B}$ since we can take $\phi = \exists x \forall y (y \leq x)$. Then $\mathcal{A} \models \neg\phi$, but $\mathcal{B} \models \phi$.

Note that in this case, the n of the definition is 0. In fact, if $\mathcal{A} \prec \mathcal{B}$ then $\mathcal{A} \equiv \mathcal{B}$ by definition using the case $n = 0$.

Example: Let $\mathcal{A} = ((0,1), \leq)$, and $\mathcal{B} = ((0,2), \leq)$. Then $\mathcal{A} \prec \mathcal{B}$. Given $a_1, \ldots, a_n$ and ϕ, there is a quantifier-free $B(v_1, \ldots, v_n)$ such that

$$\text{DLO} \models \phi(v_1, \ldots, v_n) \leftrightarrow B(v_1, \ldots, v_n).$$

So,

$$\begin{aligned} \mathcal{A} \models \phi(a_1, \ldots, a_n) &\Leftrightarrow \mathcal{A} \models B(a_1, \ldots, a_n) \\ &\Leftrightarrow \mathcal{B} \models B(a_1, \ldots, a_n) \\ &\Leftrightarrow \mathcal{B} \models \phi(a_1, \ldots, a_n). \end{aligned}$$

Example: Let $\mathcal{A} = ((0,1], \leq, \mathtt{1})$, and $\mathcal{B} = ((0,2], \leq, \mathtt{1})$. ($\mathtt{1}$ is a constant denoting the object '1'.) Then $\mathcal{A} \subseteq \mathcal{B}$, but $\mathcal{A} \not\prec \mathcal{B}$, because $\mathcal{A} \models \forall x (x \leq \mathtt{1})$, but $\mathcal{B} \not\models \forall x (x \leq \mathtt{1})$.

Example: Let $\mathcal{A} = ((0,1], \leq)$, and $\mathcal{B} = ((0,2], \leq)$. Then $\mathcal{A} \subseteq \mathcal{B}$, but $\mathcal{A} \not\prec \mathcal{B}$, because we can take $a_1 = 1$ (the object '1') and then $\mathcal{A} \models \forall x (x \leq a_1)$, but $\mathcal{B} \not\models \forall x (x \leq a_1)$. (Remember, $\forall x (x \leq a_1)$ is shorthand for $\forall x (x \leq y)[s(a_1/y)]$ for any object assignment s).

Definition: Let $\mathcal{A}$ be a structure with universe A for the language $\mathcal{L}$. Let

$$\mathcal{L}_A = \mathcal{L} \cup \{c_a : a \in A\}.$$

Then the *elementary diagram* of $\mathcal{A}$ is the set of sentences ϕ in the expanded language $\mathcal{L}_A$ such that

$$(\mathcal{A}, a)_{a \in A} \models \phi$$

where $(\mathcal{A}, a)_{a \in A}$ is the expansion of $\mathcal{A}$ to the language $\mathcal{L}_A$ such that c_a is interpreted as a; i.e., $(\mathcal{A}, a)_{a \in A}$ gives names to all of the objects in $\mathcal{A}$.

Note: the elementary diagram of $\mathcal{A}$ is complete. If Γ_A is the elementary diagram of $\mathcal{A}$, then $\Gamma_A = \mathrm{Th}((\mathcal{A}, a)_{a \in A})$.

Proposition: Let Γ_A be the elementary diagram of $\mathcal{A}$, and let $\mathcal{A} \subseteq \mathcal{B}$. Then $\mathcal{A} \prec \mathcal{B}$ iff $(\mathcal{B}, a)_{a \in A} \models \Gamma_A$.
Proof:

$$(\mathcal{B}, a)_{a \in A} \models \phi(a_1, \ldots, a_k) \Leftrightarrow \mathcal{B} \models \phi(a_1, \ldots, a_k).$$

Similarly,

$$(\mathcal{A}, a)_{a \in A} \models \phi(a_1, \ldots, a_k) \Leftrightarrow \mathcal{A} \models \phi(a_1, \ldots, a_k).$$

Then, just unwind the definitions ... □

Elementary Embeddings

Math 260B - Mathematical Logic

March 10, 1989

Definition: Let f be a 1-1 function on $|\mathcal{A}|$. Then f is an *elementary embedding* of $\mathcal{A}$ into $\mathcal{B}$ (written $f : \mathcal{A} \prec \mathcal{B}$) iff $f(\mathcal{A}) \prec \mathcal{B}$.

Notes: f being a 1-1 function on $|\mathcal{A}|$ implies that $|f(\mathcal{A})| = f(|\mathcal{A}|)$. Also, $\mathcal{A} \overset{\sim}{\prec} \mathcal{B}$ means that there is an f such that $f : \mathcal{A} \prec \mathcal{B}$.

Theorem: Let $\mathcal{F}$ be a set of elementarily equivalent structures (in some common language). Then there is a structure $\mathcal{B}$ such that every structure $\mathcal{A} \in \mathcal{F}$ is elementarily embeddable in $\mathcal{B}$.

Proof: Without loss of generality, assume that the $\mathcal{A}$'s in $\mathcal{F}$ have disjoint universes. For $\mathcal{A} \in \mathcal{F}$, let $A = |\mathcal{A}|$ and let $\mathcal{L}_A$ be the language of A plus $\{c_a : a \in A\}$. Let Γ_A be the elementary diagram of $\mathcal{A}$. Let the theory T be such that $T = \bigcup_{\mathcal{A} \in \mathcal{F}} \Gamma_A$ with language $\cup \mathcal{L}_A$.

> *Claim 1:* T is consistent.
>
> *Claim 2:* If $\mathcal{B} \models T$, then for each $\mathcal{A} \in \mathcal{F}$, $\mathcal{A} \overset{\sim}{\prec} \mathcal{B}$.
>
> *Proof of 2:* Given $\mathcal{B} \models T$, and $\mathcal{A} \in \mathcal{F}$, let $f(a) = c_a^{\mathcal{B}}$. So $f : |\mathcal{A}| \to |\mathcal{B}|$. Now f is 1-1 since for all $a, a^{'} \in |\mathcal{A}|$, if $a \neq a^{'}$, then $c_a \neq c^{'}_a \in \Gamma_A$. Also, for all formulas ϕ in the language of $\mathcal{A}$,
>
> $$\begin{aligned} \mathcal{A} \models \phi(a_1, \ldots, a_n) &\Leftrightarrow \mathcal{B} \models \phi(c_{a_1}, \ldots, c_{a_n}) && (1)\\ &\Leftrightarrow \mathcal{B} \models \phi(f(a_1), \ldots, f(a_n)). && (2) \end{aligned}$$
>
> (1) is because $\mathcal{B} \models \Gamma_A$, and (2) is because $f(a_i) = c_{a_i}$. So f is an isomorphism of $\mathcal{A}$ onto the substructure of $\mathcal{B}$ with universe $f(|\mathcal{A}|)$, and this substructure is an elementary substructure. I.e.,

$$
\begin{aligned}
\mathcal{A} \models \phi(a_1, \ldots, a_n) & & (3)\\
&\Leftrightarrow \quad \mathcal{B} \models \phi(c_{a_1}, \ldots, c_{a_n}) & (4)\\
&\Leftrightarrow \quad f(\mathcal{A}) \models \phi(c_{a_1}, \ldots, c_{a_n}). & (5)
\end{aligned}
$$

(4) $\Leftrightarrow$ (5) means that $f(\mathcal{A}) \prec \mathcal{B}$, and (3) $\Leftrightarrow$ (5) means that $\mathcal{A} \cong f(\mathcal{A})$. □

Proof of 1: By compactness, it suffices to show that any finite subset is consistent. Suppose that $S = \{\phi_1, \ldots, \phi_n\}$. By taking conjunctions of ϕ's as necessary, without loss of generality each $\phi_i \in \Gamma_{A_i}$ for distinct A_i's. (I.e., if two ϕ's come from the same A_i, then conjoin them.)

$$S = \{\psi_1(c_{a_{1,1}}, \ldots, c_{a_{1,k}}), \ldots, \psi_n(c_{a_{n,1}}, \ldots, c_{a_{n,k}})\} \qquad (6)$$

where the ψ_i's are in the original language and the $a_{i,j}$'s are in A_i. So S is consistent iff

$$\exists x_{1,1} \ldots \exists x_{n,k}(\psi_1(x_{1,1}, \ldots, x_{1,k}) \wedge \ldots \wedge \psi_n(x_{n,1}, \ldots, x_{n,k})) \quad (7)$$

is satisfiable. Note that (6) is just a skolemization of (7). By prenex operations, (7) is equivalent to

$$\exists x_{1,1} \ldots \exists x_{1,k}\psi_1(\vec{x}) \wedge \ldots \wedge \exists x_{n,1} \ldots \exists x_{n,k}\psi_n(\vec{x}). \qquad (8)$$

Note that each conjunct in (8) for ψ_i is true in $\mathcal{A}_i$. Hence each conjunct is true in all $\mathcal{A} \in \mathcal{F}$ since all such $\mathcal{A}$'s are elementarily equivalent. So any $\mathcal{A} \in \mathcal{F}$ is a model of (8) and (8) is consistent. Since n was arbitrary, the claim and the theorem are established. □

Example: Let $\mathcal{F} = \{\mathcal{A}_1, \mathcal{A}_2\}$ where

$$
\begin{aligned}
\mathcal{A}_1 &= ((0,1) \cup (Q \cap (1,2)), \leq), \text{ and}\\
\mathcal{A}_2 &= (Q \cap (0,1) \cup (1,2), \leq).
\end{aligned}
$$

Then

$$\mathcal{B} \;=\; ((0,1) \cup (1,2), \leq)$$

is such that $\mathcal{A}_1$ and $\mathcal{A}_2$ are elementarily embedded into $\mathcal{B}$.

Elementary Substructures

Math 260B - Mathematical Logic

March 13, 1989

Sometimes it is hard to show that one structure is an elementary substructure of another structure by showing that they satisfy the same set of formulas. The following theorem presents another criterion for being an elementary substructure.

Theorem: $\mathcal{A} \prec \mathcal{B}$ iff

1. $\mathcal{A} \subseteq \mathcal{B}$, and
2. for every formula $\phi(x_1, \ldots, x_{n+1})$ and every n-tuple $a_1, \ldots, a_n \in |\mathcal{A}|$, if

$$\mathcal{B} \models \exists x_{n+1} \phi(a_1, \ldots, a_n, x_{n+1}),$$

then there is an $a_{n+1} \in |\mathcal{A}|$ such that

$$\mathcal{B} \models \phi(a_1, \ldots, a_{n+1}).$$

Proof: ($\Rightarrow$) Easy. If

$$\mathcal{B} \models \exists x_{n+1} \phi(a_1, \ldots, a_n, x_{n+1}),$$

then so does $\mathcal{A}$ since $\mathcal{A} \prec \mathcal{B}$. So there is such an a_{n+1}.
($\Leftarrow$) By induction on the complexity of ψ, we show that for all $a_1, \ldots, a_m$,

$$\mathcal{A} \models \psi(a_1, \ldots, a_m) \Leftrightarrow \mathcal{B} \models \psi(a_1, \ldots, a_m),$$

(and so $\mathcal{A} \prec \mathcal{B}$.)
Basis: ψ is atomic. Since $\mathcal{A} \subseteq \mathcal{B}$,

(i) if $t(x_1,\ldots,x_m)$ is a term, then $t(a_1,\ldots,a_m)^{\mathcal{A}} = t(a_1,\ldots,a_m)^{\mathcal{B}}$, and

(ii) relations agree on elements in $|\mathcal{A}|$.

(Note: in this case, we only needed the fact that $\mathcal{A} \subseteq \mathcal{B}$.)
Induction: Cases where the outer connective of ψ is $\wedge$, $\vee$, $\neg$, or $\rightarrow$ are easy using the induction hypothesis. And since $\forall x$ can be written as $\neg\exists x\neg$, it suffices to consider ψ of the form $\exists x\psi^{'}(x_1,\ldots,x_m,x)$. Let $a_1,\ldots,a_m \in |\mathcal{A}|$. Then

$$\begin{aligned}
&\mathcal{A} \models \psi(a_1,\ldots,a_m) && (1)\\
&\quad\Leftrightarrow\quad \exists a_{m+1} \in |\mathcal{A}| \text{ such that } \mathcal{A} \models \psi^{'}(a_1,\ldots,a_m,a_{m+1}) && (2)\\
&\quad\Leftrightarrow\quad \exists a_{m+1} \in |\mathcal{A}| \text{ such that } \mathcal{B} \models \psi^{'}(a_1,\ldots,a_m,a_{m+1}) && (3)\\
&\quad\Leftrightarrow\quad \exists a_{m+1} \in |\mathcal{B}| \text{ such that } \mathcal{B} \models \psi^{'}(a_1,\ldots,a_m,a_{m+1}) && (4)\\
&\quad\Leftrightarrow\quad \mathcal{B} \models \psi(a_1,\ldots,a_m). && (5)
\end{aligned}$$

(1) $\Leftrightarrow$ (2) by the definition of truth, (2) $\Leftrightarrow$ (3) by the induction hypothesis, (3) $\Rightarrow$ (4) since $|\mathcal{A}| \subseteq |\mathcal{B}|$, (4) $\Rightarrow$ (3) by the second condition in the theorem, and (4) $\Leftrightarrow$ (5) by the definition of truth. □

An application of this theorem is the

Downward-Löwenheim-Skolem Theorem: (strong version) Suppose that $\mathcal{A}$ is a structure in the language $\mathcal{L}$. Let $||\mathcal{L}|| = \alpha$. Let $X \subseteq |\mathcal{A}|$ with $||X|| = \beta \geq \alpha$. (Note that β is infinite by definition of $||\ ||$.) Then there is an elementary substructure $\mathcal{B}$ of $\mathcal{A}$ such that

1. $X \subseteq |\mathcal{B}|$, and

2. $\mathcal{B}$ has cardinality β.

Proof: (To help understand the proof, keep in mind $\beta = \omega$, but note that the proof works for larger β.) Given a set $Y \subseteq |\mathcal{A}|$, construct $Y^{'}$ as follows:

```
foreach $a_1,\ldots,a_m \in Y$
    foreach formula $\exists x_{m+1}\phi(x_1,\ldots,x_{m+1})$
        if $\mathcal{A} \models \exists x_{m+1}\phi(a_1,\ldots,a_m,x_{m+1})$
          choose some $a \in \mathcal{A}$ such that $\mathcal{A} \models \phi(a_1,\ldots,a_m,a)$
        end if
    end foreach
end foreach
```

Let $Y^{'}$ be Y plus all the a's chosen. Note that $||Y^{'}|| = \beta$ if $||Y|| = \beta$. Now let $X_0 = X$, let $X_{i+1} = X_i^{'}$ as in the above construction, and let $B = \bigcup_{i \in N} X_i$. Then $\mathcal{B}$ is the substructure of $\mathcal{A}$ with universe B.

> *Claim:* Every constant symbol $c \in \mathcal{L}$ has $c^{\mathcal{A}} \in \mathcal{B}$.
>
> *Proof*: Since $\mathcal{A} \models \exists x(x = c)$, $c^{\mathcal{A}} \in X$. $\square$
>
> *Claim:* $\mathcal{B}$ is closed under function symbols.
>
> *Proof*: Let f be a k-ary function symbol. If $a_1, \ldots, a_k \in |\mathcal{B}|$, then $a_1, \ldots, a_k \in X_r$ for some r. Since $\mathcal{A} \models \exists x(x = f(a_1, \ldots, x_k))$, $f^{\mathcal{A}}(a_1, \ldots, a_k) \in X_{r+1} \subseteq B$. $\square$

The above claims show that $\mathcal{B} \subseteq \mathcal{A}$. Now we need

> *Claim:* If $a_1, \ldots, a_k \in B$ and $\mathcal{A} \models \exists x \phi(a_1, \ldots, a_k, x)$, then $\exists a \in B$ such that $\mathcal{A} \models \phi(a_1, \ldots, a_k, a)$.
>
> *Proof*: If again $a_1, \ldots, a_k \in X_r$ for some r, then $\exists a \in X_{r+1}$ such that $\mathcal{A} \models \phi(a_1, \ldots, a_m, a)$. $\square$

Example: DLO in the language $\leq, c, d$. Let

$$\begin{aligned} \mathcal{A} &= ((0,1) \cup ((1,2) \cap Q), \leq, \texttt{1/2}, \texttt{3/2}) \\ \mathcal{B} &= (((0,1) \cap Q) \cup (1,2), \leq, \texttt{1/2}, \texttt{3/2}) \end{aligned}$$

Then $\mathcal{A} \equiv \mathcal{B}$, but $\mathcal{A} \overset{\sim}{\not\prec} \mathcal{B}$ and $\mathcal{B} \overset{\sim}{\not\prec} \mathcal{A}$ because in $\mathcal{A}$ there are uncountably many objects less than 1/2 but only countably many such objects in $\mathcal{B}$, and in $\mathcal{A}$ there are countably many objects greater than 3/2 but uncountably many such objects in $\mathcal{B}$.

Model-Completeness, Weak Elimination of Quantifiers

Math 260C - Mathematical Logic

April 3, 1989

Definition: A theory T is *model-complete* iff for any two models $\mathcal{A}$ and $\mathcal{B}$ of T, if $\mathcal{A} \subseteq \mathcal{B}$, then $\mathcal{A} \prec \mathcal{B}$.

Example: DLO without endpoints is model-complete.

Proof: Suppose that $\mathcal{A} \subseteq \mathcal{B}$ and that $\mathcal{A}, \mathcal{B} \models$ DLO. Let $\phi(x_1, \ldots, x_n)$ be a formula (with $n > 0$),[1] and let $a_1, \ldots, a_n \in |\mathcal{A}|$. In order to show that $\mathcal{A} \prec \mathcal{B}$, we want to show that $\mathcal{A} \models \phi(a_1, \ldots, a_n)$ iff $\mathcal{B} \models \phi(a_1, \ldots, a_n)$. By elimination of quantifiers for DLO, there is a quantifier-free formula $\psi(x_1, \ldots, x_n)$ such that DLO $\models \phi \leftrightarrow \psi$. So

$$\begin{array}{rcll} \mathcal{A} \models \phi(a_1, \ldots, a_n) & \Leftrightarrow & \mathcal{A} \models \psi(a_1, \ldots, a_n) & \text{since } \mathcal{A} \models \text{DLO} \\ & \Leftrightarrow & \mathcal{B} \models \psi(a_1, \ldots, a_n) & \text{since } \mathcal{A} \subseteq \mathcal{B}, \text{ and} \\ & & & \psi \text{ is quantifier-free} \\ & \Leftrightarrow & \mathcal{B} \models \phi(a_1, \ldots, a_n) & \text{since } \mathcal{B} \models \text{DLO. } \Box \end{array}$$

Fact: Q is not model-complete. This is proved by noting (1) that every model of Q contains a substructure isomorphic to the standard integers and (2) that Q is not complete.

Example: $T = \text{Th}(N, S, \leq)$ is not model-complete.

Proof: Let $\mathcal{A} = (N, S, \leq)$, and let $\mathcal{C} = (\{-1, 0, 1, 2, \ldots\}, S, \leq)$. $\mathcal{A} \equiv \mathcal{C}$, so $\mathcal{C} \models T$. Clearly, $\mathcal{A} \subseteq \mathcal{C}$. But $\mathcal{A} \models \forall y(0 \leq y)$ while $\mathcal{C} \not\models \forall y(0 \leq y)$. So $\mathcal{A} \not\prec \mathcal{C}$. $\Box$

[1] In using elimination of quantifiers, if $n > 0$, then a formula equivalent to ϕ will not have more free variables than ϕ has; if $n = 0$, then the equivalent formula would have 1 free variable. In justifying the case for $n = 0$, note that we could always add $\wedge x = x$ to ϕ.

This example doesn't work for $\text{Th}(N, 0, S, \leq)$ since in order for $\mathcal{A} \subseteq \mathcal{C}$, the interpretation of 0 must be the same in both $\mathcal{A}$ and $\mathcal{C}$.[2] In fact, $\text{Th}(N, 0, S, \leq)$ admits elimination of quantifiers, so a proof similar to the one used in the first example would work here as well.

Example: Let RCF be real closed fields. The non-logical symbols are $0, 1, +$, and $*$. RCF contains the following axioms.

- The field axioms.
- For every odd degree polynomial, an axiom stating that it has a root.
- For every degree ≥ 3, an axiom stating that a polynomial of that degree can be factored into a linear and square parts; i.e., for degree 3,

$$\begin{aligned} & \forall a_1, \ldots, a_4 \exists b_1, b_2, c_1, c_2, c_3((b_1 x + b_2)(c_1 x^2 + c_2 x + c_3) \\ = \quad & a_1 x^3 + a_2 x^2 + a_3 x + a_4). \end{aligned}$$

- Every object has a square root; i.e.,

$$\forall x \exists y (y * y = x \vee y * y + x = 0).$$

- Zero is not the sum of squares; i.e.,

$$\forall x_1 \ldots \forall x_k (x_1 \neq 0 \rightarrow 0 \neq x_1^2 + x_2^2 + \ldots x_k^2).$$

RCF $= \text{Th}(R, 0, 1, +, *)$ is both complete and model-complete.

Example: Let ACF be algebraically closed fields. It contains the same non-logical symbols as RCF, the field axioms, and, for each degree, an axiom stating that every polynomial of that degree has a root; i.e.,

$$\forall a_1, \ldots, a_n \exists b (a_1 b^n + a_2 b^{n-1} + \ldots + a_{n+1} = 0).$$

ACF is not complete but is model-complete.

[2] I.e., $0^{\mathcal{A}} = 0^{\mathcal{C}}$. But then $\mathcal{C} \not\models T$, since $\mathcal{C} \not\models \forall x (0 \neq S(x))$.

Example: Let ACF_p be ACF plus

$$\underbrace{1+1+\ldots+1}_{p}=0,$$

for p a prime. ACF_p is both complete and model-complete.

Example: Let ACF_0 be ACF plus $\{1+1+\ldots+1 \neq 0\}$. ACF_0 is both complete and model-complete.

Usually, in order to show that a theory is model-complete, you have to know a lot about the theory. Showing that a model is model-complete is therefore a good indication that the theory is well understood.

Definition: A theory T *admits weak elimination of quantifiers* iff for any formula ϕ, there is a universal formula ψ such that $T \models \phi \leftrightarrow \psi$.

Theorem: If T admits weak elimination of quantifiers, then for all ϕ there is an existential formula ψ such that $T \models \phi \leftrightarrow \psi$.

Proof: Given ϕ, there is a universal formula ψ such that $T \models (\neg\phi) \leftrightarrow \psi$. So $T \models \phi \leftrightarrow \neg\psi$. $\neg\psi$ can be converted to existential form. □

Theorem: If T admits weak elimination of quantifiers, then T is model-complete.

Proof: Next time.

Fact: The converse also holds.

Weak Elimination of Quantifiers, Preservation

Math 260C - Mathematical Logic

April 5, 1989

Theorem: If a theory T admits weak elimination of quantifiers, then T is model-complete.

> **Lemma 1**: If $\phi(x_1, \ldots, x_k)$ is a universal formula, $\mathcal{A} \subseteq \mathcal{B}$, and $a_1, \ldots, a_k \in |\mathcal{A}|$, then if $\mathcal{B} \models \phi(a_1, \ldots, a_k)$ then $\mathcal{A} \models \phi(a_1, \ldots, a_k)$. (I.e., universal formulas are preserved under substructures.)
>
> *Proof*: Suppose ϕ is $\forall y_1 \ldots \forall y_\ell \psi(x_1, \ldots, x_k, y_1, \ldots, y_\ell)$. Then,
>
> $$\begin{array}{lcll} \mathcal{B} \models \phi(\vec{a}) & \Leftrightarrow & \forall b_1 \ldots \forall b_\ell \in \mathcal{B},\ \mathcal{B} \models \psi(\vec{a}, \vec{b}) & \text{by defn. of truth} \\ & \Rightarrow & \forall b_1 \ldots \forall b_\ell \in \mathcal{A},\ \mathcal{B} \models \psi(\vec{a}, \vec{b}) & \text{since } |\mathcal{A}| \subseteq |\mathcal{B}| \\ & \Leftrightarrow & \forall b_1 \ldots \forall b_\ell \in \mathcal{A},\ \mathcal{A} \models \psi(\vec{a}, \vec{b}) & \text{since } \mathcal{A} \subseteq \mathcal{B} \\ & \Leftrightarrow & \mathcal{A} \models \phi(\vec{a}). & \text{by defn. of truth. } \square \end{array}$$
>
> **Lemma 2**: If $\phi(x_1, \ldots, x_k)$ is an existential formula, $\mathcal{A} \subseteq \mathcal{B}$, and $a_1, \ldots, a_k \in |\mathcal{A}|$, then if $\mathcal{A} \models \phi(a_1, \ldots, a_k)$ then $\mathcal{B} \models \phi(a_1, \ldots, a_k)$. (I.e., existential formulas are preserved under superstructures.)
>
> *Proof*: This can be proved in a similar way that lemma 1 was proved, or it can be noted that lemma 2 is just the contrapositive of lemma 1. $\square$

Proof of theorem: Given $\mathcal{A}, \mathcal{B} \models T$, and $\mathcal{A} \subseteq \mathcal{B}$, we will show that for $\phi(x_1, \ldots, x_k)$ and $a_1, \ldots, a_k \in |\mathcal{A}|$,

$$\mathcal{A} \models \phi(a_1, \ldots, a_k) \Leftrightarrow \mathcal{B} \models \phi(a_1, \ldots, a_k).$$

By weak elimination of quantifiers, there is a universal formula $\psi(x_1, \ldots, x_k)$ such that $T \models \phi \leftrightarrow \psi$, and an existential formula $\chi(x_1, \ldots, x_k)$ such that $T \models \phi \leftrightarrow \chi$. Now,

$$\begin{array}{rcll}
\mathcal{A} \models \phi(\vec{a}) & \Leftrightarrow & \mathcal{A} \models \chi(\vec{a}) & \text{since } \mathcal{A} \models T, \text{ and } T \models \phi \leftrightarrow \chi \\
& \Rightarrow & \mathcal{B} \models \chi(\vec{a}) & \text{by lemma 2} \\
& \Leftrightarrow & \mathcal{B} \models \phi(\vec{a}) & \text{since } \mathcal{B} \models T, \text{ and } T \models \phi \leftrightarrow \chi \\
& \Leftrightarrow & \mathcal{B} \models \psi(\vec{a}) & \text{since } \mathcal{B} \models T, \text{ and } T \models \phi \leftrightarrow \psi \\
& \Rightarrow & \mathcal{A} \models \psi(\vec{a}) & \text{by lemma 1} \\
& \Leftrightarrow & \mathcal{A} \models \phi(\vec{a}) & \text{since } \mathcal{A} \models T, \text{ and } T \models \phi \leftrightarrow \chi. \ \Box
\end{array}$$

Theorem: If a theory is model-complete, then it admits weak elimination of quantifiers.

Proof: In Chang & Keisler, proposition 3.1.7.

Preservation

Definition: A *chain of structures* is a sequence of structures

$$\mathcal{A}_0 \subseteq \mathcal{A}_1 \subseteq \mathcal{A}_2 \subseteq \ldots.$$

Fact: In a chain of structures, $\mathcal{A}_i \subseteq \mathcal{A}_j$ for $i < j$.

Definition: $\mathcal{B} = \bigcup_{i \in N} \mathcal{A}_i$ is the structure with

- domain $\bigcup_i |\mathcal{A}_i|$,
- constants $c^{\mathcal{B}} = c^{\mathcal{A}_i}$,
- functions $f^{\mathcal{B}}$ which extend $f^{\mathcal{A}_i}$ for all i (or, thinking of functions as ordered tuples, $f^{\mathcal{B}} = \bigcup_i f^{\mathcal{A}_i}$), and
- predicates $P^{\mathcal{B}} = \bigcup_i P^{\mathcal{A}_i}$.

Definition: An *elementary chain of structures* is a sequence

$$\mathcal{A}_0 \prec \mathcal{A}_1 \prec \mathcal{A}_2 \prec \dots .$$

Fact: In an elementary chain of structures, $\mathcal{A}_i \prec \mathcal{A}_j$ for $i < j$.

Elementary Chain Theorem: $\mathcal{A}_n \prec \bigcup_i \mathcal{A}_i$ for all n.

Proof: Let $\mathcal{B} = \bigcup_i \mathcal{A}_i$. Then, given $\phi(x_1, \dots, x_k)$ and $a_1, \dots, a_k \in |\mathcal{A}_n|$, we need to show that

$$\mathcal{A}_n \models \phi(a_1, \dots, a_k) \Leftrightarrow \mathcal{B} \models \phi(a_1, \dots, a_k). \tag{1}$$

The proof is by induction on the complexity of ϕ (letting n and $a_1, \dots, a_k$ vary).

Basis: ϕ is atomic. Then (1) holds trivially, since $\mathcal{A}_n \subseteq \mathcal{B}$.

Induction: If the outer connective of ϕ is one of $\neg, \wedge, \vee$, or $\rightarrow$, then (1) holds trivially by the induction hypothesis. Since $\forall x$ means $\neg\exists\neg x$, it suffices to consider ϕ of the form $\exists x \psi(x_1, \dots, x_k, x)$. Now,

$$\begin{array}{lcll}
\mathcal{B} \models \exists x \psi(\vec{a}, x) & \Leftrightarrow & \exists a \in |\mathcal{B}|,\ \mathcal{B} \models \psi(\vec{a}, a) & \text{by defn. of truth} \\
& \Leftrightarrow & (\exists m \geq n) \exists a \in |\mathcal{A}_m|,\ \mathcal{B} \models \psi(\vec{a}, a) & \text{by defn. of } \mathcal{B} \\
& \Leftrightarrow & (\exists m \geq n) \exists a \in |\mathcal{A}_m|,\ \mathcal{A}_m \models \psi(\vec{a}, a) & \text{by ind. hyp.} \\
& \Leftrightarrow & (\exists m \geq n) \mathcal{A}_m \models \exists x \psi(\vec{a}, x) & \text{by defn. of truth} \\
& \Leftrightarrow & (\exists m \geq n) \mathcal{A}_n \models \exists x \psi(\vec{a}, x) & \text{since } \mathcal{A}_n \prec \mathcal{A}_m \\
& \Leftrightarrow & \mathcal{A}_n \models \exists x \psi(\vec{a}, x) & \text{since } m \text{ isn't bound. } \square
\end{array}$$

Definition: A theory T is *preserved under substructures* iff if $\mathcal{A} \models T$ and $\mathcal{B} \subseteq \mathcal{A}$, then $\mathcal{B} \models T$.

Definition: A theory T is *preserved under unions of chains* iff if $\mathcal{A}_0 \subseteq \mathcal{A}_1 \subseteq \dots$ is a chain, and $\mathcal{A}_n \models T$ for all n, then $\bigcup_i \mathcal{A}_i \models T$.

Theorem: Any theory T is preserved under unions of elementary chains.

Proof: If $\mathcal{A} \prec \mathcal{B}$, then $\mathcal{A} \equiv \mathcal{B}$. So $\mathcal{A} \models T$ iff $\mathcal{B} \models T$. $\square$

Homework: Find a theory that is not preserved under unions of chains.

Preservation Theorems

Math 260C - Mathematical Logic

April 7, 1989

Homework Answer 1: DLO with left and right endpoints and with $\leq$ as the only nonlogical symbol is not preserved under unions of chains. Let $\mathcal{A}_i = ([-(i+1),(i+1)], \leq)$. Then $\bigcup_i \mathcal{A}_i = (R, \leq)$. So $\mathcal{A}_i \subseteq \mathcal{A}_{i+1}$ for all i. But $\bigcup_i \mathcal{A}_i \not\equiv \mathcal{A}_n$ for any n. This example is a proof of the following

Theorem: There is a chain of elementarily equivalent structures whose union is not elementarily equivalent to elements of the chain.

Homework Answer 2: Let $\mathcal{A}_i = (\{0,\ldots,i\}, \leq)$, and let the theory state that there is a maximal element; i.e., $\exists x \forall y (y \leq x)$.

Definition: Let $\mathcal{A}$ and $\mathcal{B}$ be structures. $h : |\mathcal{A}| \overset{\text{onto}}{\rightarrow} |\mathcal{B}|$ is a homomorphism iff

1. $h(c^{\mathcal{A}}) = c^{\mathcal{B}}$ for all constant symbols c,
2. if $P^{\mathcal{A}}(a_1,\ldots,a_n)$ then $P^{\mathcal{B}}(h(a_1),\ldots,h(a_n))$ for all predicate symbols P and all $a_1,\ldots,a_n \in |\mathcal{A}|$,
3. $h(f^{\mathcal{A}}(a_1,\ldots,a_n)) = f^{\mathcal{B}}(h(a_1),\ldots,h(a_n))$ for all function symbols f and all $a_1,\ldots,a_n \in |\mathcal{A}|$.

Example: Note that h is not necessarily 1-1. As a trivial example, $h : (\{0,1,2,3,4\}, \leq) \rightarrow (\{0\}, \leq)$. Note also that homomorphisms don't necessarily preserve sentences; e.g. $\exists x \exists y (x \neq y)$ is not preserved.

Definition: A theory T is preserved under homomorphisms iff whenever $\mathcal{A} \models T$, and $\mathcal{B}$ is the homomorphic image of $\mathcal{A}$, then $\mathcal{B} \models T$.

Definition: A Π_1 sentence is a universal sentence. A Π_2 sentence is of the form $\forall x_1 \ldots \forall x_k \exists y_1 \ldots \exists y_\ell \phi$ with ϕ quantifier-free. A *positive* sentence is one with no occurrences of $\neg$ or $\rightarrow$.

Theorems: A theory T is preserved under

1. substructures iff T has a set of Π_1 axioms,
2. unions of chains iff T has a set of Π_2 axioms, and
3. homomorphisms iff T has a set of positive axioms.

Example: Horn clauses are not preserved under homomorphisms. Take $\leftarrow P(0)$; i.e., $\neg P(0)$. Then $\mathcal{A} = (\{0\}, \emptyset) \models \neg P(0)$, and $\mathcal{B} = (\{0\}, \{0\}) \not\models \neg P(0)$. But there is a homomorphism $h : \mathcal{A} \rightarrow \mathcal{B}$.

Example: DLO without endpoints is preserved under unions of chains but not under substructures and homomorphisms. Preservation under unions of chains is because the axioms are Π_2. As a counter-example for substructures, take $((0,1), \leq) \supseteq (\{1/2\}, \leq)$. As a counter-example for homomorphisms, take $h : ((0,1), \leq) \rightarrow (\{1/2\}, \leq)$.

Example: DLO with right and left endpoints is not preserved under substructures, unions of chains, and homomorphisms. As a counter-example for substructures, take $([0,1], \leq) \supseteq ((0,1), \leq)$. A counter-example for unions of chains is homework answer 1. As a counter-example for homomorphisms, take $\mathcal{A} = ([0,1], \leq)$, $\mathcal{B} = (\{0,1\}, \leq)$, and $h : \mathcal{A} \rightarrow \mathcal{B}$ to be

$$h(x) = \begin{cases} 0 & x \leq 1/2 \\ 1 & x > 1/2. \end{cases}$$

Example: Groups are preserved under unions of chains and homomorphisms, but not under substructures. As a counter-example for substructures, take $(Z, 0, +) \supseteq (N, 0, +)$. Note that if we had an inverse function symbol, then groups would be preserved under substructures.

Theorem: If a theory T has a set of universal axioms, then T is preserved under substructures.

Proof: Let Γ be a set of universal axioms for T. Suppose that $\mathcal{A} \models T$ and that $\mathcal{B} \subseteq \mathcal{A}$. Then $\mathcal{A} \models \phi$ for $\phi \in \Gamma$. Also, since the ϕ's are universal and by an earlier theorem, $\mathcal{B} \models \phi$ for $\phi \in \Gamma$. So $\mathcal{B} \models T$ since Γ is a set of axioms. $\Box$

Theorem: If T has a set of Π_2 axioms, then T is preserved under unions of chains.

Proof: It suffices to show that if ϕ is a Π_2 sentence, and

$$\mathcal{A}_1 \subseteq \mathcal{A}_2 \subseteq \mathcal{A}_3 \subseteq \dots$$

is a chain with $\mathcal{A}_i \models \phi$ for all i, then

$$\mathcal{B} = \bigcup_i \mathcal{A}_i \models \phi.$$

Let ϕ be

$$\forall x_1 \dots \forall x_k \exists y_1 \dots \exists y_\ell \psi(\vec{x}, \vec{y}).$$

Then we want to show that for any $a_1, \dots a_k \in |\mathcal{B}|$,

$$\mathcal{B} \models \exists y_1 \dots \exists y_\ell \psi(\vec{a}, \vec{y}),$$

and hence, by the definition of truth, $\mathcal{B} \models \phi$. Pick n large enough so that $a_1, \dots, a_k \in |\mathcal{A}_n|$. Now,

$$\mathcal{A}_n \models \exists y_1 \dots \exists y_\ell \psi(\vec{a}, \vec{y}).$$

So there are $b_1, \dots, b_\ell \in |\mathcal{A}_n|$ such that $\mathcal{A}_n \models \psi(\vec{a}, \vec{b})$. Since $\mathcal{A}_n \subseteq \mathcal{B}$, $\mathcal{B} \models \psi(\vec{a}, \vec{b})$. Hence, by the definition of truth,

$$\mathcal{B} \models \exists y_1 \dots \exists y_\ell \psi(\vec{a}, \vec{y}).$$

Since the a's were arbitrary,

$$\mathcal{B} \models \forall x_1 \dots \forall x_k \exists y_1 \dots \exists y_\ell \psi(\vec{x}, \vec{y}). \ \Box$$

Preservation Theorems (cont.)

Math 260C - Mathematical Logic

April 10, 1989

Lemma: Let T be a consistent theory. Let Δ be a set of sentences closed under finite disjunction. Then the following are equivalent.

1. T has a set of axioms which is a subset of Δ.
2. If $\mathcal{A} \models T$ and if for every $\phi \in \Delta$, $\mathcal{A} \models \phi \Rightarrow \mathcal{B} \models \phi$, then $\mathcal{B} \models T$.

Proof: $1 \Rightarrow 2$: Easy.[1]
$2 \Rightarrow 1$: Δ could be inconsistent, so let

$$\Gamma = \{\phi \in \Delta : T \models \phi\}$$

be the consistent subset of Δ implied by T.[2] Suppose that $\mathcal{B} \models \Gamma$. We want to show that $\mathcal{B} \models T$. It will then follow from the definition of logical implication that Γ is a set of axioms for T. Let

$$\Sigma = \{\neg\delta : \mathcal{B} \models \neg\delta, \delta \in \Delta\};$$

i.e., the sentences in Δ that $\mathcal{B}$ doesn't satisfy.

> *Claim:* $T \cup \Sigma$ is consistent.
>
> *Proof*: Suppose not. Then, by the compactness theorem, there is a finite subset $\{\neg\delta_1, \ldots, \neg\delta_n\}$ of Σ such that $T \cup \{\neg\delta_1, \ldots, \neg\delta_n\}$ is inconsistent. So $T \models \delta_1 \vee \ldots \vee \delta_n$. Thus $\delta_1 \vee \ldots \vee \delta_n \in \Delta$ since Δ is closed under disjunction, and so $\delta_1 \vee \ldots \vee \delta_n \in \Gamma$ by definition. So $\mathcal{B} \models \delta_1 \vee \ldots \vee \delta_n$. But $\mathcal{B} \models \neg\delta_i$ for $i = 1, \ldots, n$, which is a contradiction. $\square$

[1] Let Γ be a set of axioms for T with $\Gamma \subseteq \Delta$. Since $\mathcal{A} \models T$, $\mathcal{A} \models \Gamma$. So by 2, $\mathcal{B} \models \Gamma$. Since Γ is a set of axioms for T, $\mathcal{B} \models T$.

[2] If T has a set of axioms from Δ, then Γ is such a set.

Since $T \cup \Sigma$ is consistent, it has a model. So let $\mathcal{A} \models T \cup \Sigma$. Now $\mathcal{A} \models T$, and for all $\phi \in \Delta$, if $\mathcal{A} \models \phi$ then $\mathcal{B} \models \phi$ by definition of Σ.[3] So by 2, $\mathcal{B} \models T$. □

Theorem: If T is finitely axiomatizable and if Π is a set of axioms for T, then there is a finite subset $\Pi^{'}$ of Π which is a set of axioms for T.

Proof: Since T is finitely axiomatizable, let Γ be a finite set of axioms for T. Since Π is a set of axioms for T, $\Pi \models \Gamma$. By the compactness theorem, there is a finite $\Pi^{'} \subseteq \Pi$ such that $\Pi^{'} \models \Gamma$. □

The above lemma will be very useful for proving the preservation theorems. In the proofs of those theorems, we will let Δ be a certain set of formulas and then prove 2.

Definition: Let $\mathcal{B}$ be a structure. The set of atomic and negated atomic sentences true in $(\mathcal{B}, b)_{b \in |\mathcal{B}|}$ is called the *atomic diagram* of $\mathcal{B}$. I.e., if

$$\Delta_B = \{\text{atomic or negated atomic } \psi(c_{b_1}, \ldots, c_{b_n}) : \mathcal{B} \models \psi(b_1, \ldots, b_n)\}$$

where the c_b's are new constant symbols for elements in $|\mathcal{B}|$, then Δ_B is the atomic diagram of $\mathcal{B}$. In particular, if ψ is quantifier free (not necessarily atomic or negated atomic) and $\mathcal{B} \models \psi(b_1, \ldots, b_n)$, then $\Delta_B \models \psi(c_{b_1}, \ldots, c_{b_n})$.

Theorem: A theory T is preserved under substructures iff T has a set of universal axioms.

Proof: $\Leftarrow$: we did last time.
$\Rightarrow$: Let Δ be the set of sentences logically equivalent to the set of universal sentences.[4] Suppose that $\mathcal{A} \models T$ and that for all $\phi \in \Delta$, $\mathcal{A} \models \phi \Rightarrow \mathcal{B} \models \phi$. We want to show that $\mathcal{B} \models T$ so that we can invoke the above lemma to prove the theorem. Every universal sentence true in $\mathcal{A}$ is also true in $\mathcal{B}$, so every existential sentence true in $\mathcal{B}$ is also true in $\mathcal{A}$. Let L be the language of T, and let L_B be L plus new constant symbols c_b for $b \in |\mathcal{B}|$. Let Δ_B be the atomic diagram of $\mathcal{B}$.

[3] I.e., if $\mathcal{B} \not\models \phi$, then $\mathcal{B} \models \neg\phi$, and so $\neg\phi \in \Sigma$. But $\mathcal{A} \models \Sigma$, hence $\mathcal{A} \models \neg\phi$, a contradiction.

[4] We can't just let Δ be *the* set of universal sentence because we want to use the lemma, and the lemma requires Δ to be closed under finite disjunction.

> *Claim:* $T^{'} = T \cup \Delta_B$ is consistent.
>
> *Proof*: Suppose not. Then by the compactness theorem, there is a finite set
>
> $$\{\psi_1(c_{b_1}, \ldots, c_{b_n}), \ldots, \psi_k(c_{b_1}, \ldots, c_{b_n})\} \subseteq \Delta_B$$
>
> which is inconsistent with T. So (inverse skolemizing),[5]
>
> $$T^{''} = T \cup \exists x_1 \ldots \exists x_n(\psi_1(x_1, \ldots, x_n) \wedge \ldots \wedge \psi_k(x_1, \ldots, x_n))$$
>
> is inconsistent since the c_b's don't occur in T (they're new constant symbols). But,
>
> $$\mathcal{B} \models \exists \vec{x}(\psi_1(\vec{x}) \wedge \ldots \wedge \psi_k(\vec{x}))$$
>
> by definition of Δ_B and letting $x_1, \ldots, x_n = b_1, \ldots, b_n$. So
>
> $$\mathcal{A} \models \exists \vec{x}(\psi_1(\vec{x}) \wedge \ldots \wedge \psi_k(\vec{x}))$$
>
> since the formula is existential. Then, since $\mathcal{A} \models T$, $\mathcal{A} \models T^{''}$, which is a contradiction since $T^{''}$ is inconsistent. □

Since $T \cup \Delta_B$ is consistent, it has a model. So let $\mathcal{C}^+ \models T \cup \Delta_B$. Let $\mathcal{C}$ be the model obtained by restricting $\mathcal{C}^+$ to the language L of T. By taking an isomorphic copy of $\mathcal{C}^+$ if necessary, we can assume without loss of generality that $c_b^{\mathcal{C}^+} = b$. So $\mathcal{B} \subseteq \mathcal{C}$ because $\mathcal{C}^+ \models \Delta_B$. And $\mathcal{C} \models T$ because $\mathcal{C}^+ \models T$ and T doesn't involve any of the new constant symbols. So $\mathcal{B} \models T$ because T is assumed to be preserved under substructures. □

Homework: Prove that T is preserved under extensions (opposite of substructures) iff T has a set of existential axioms.

[5] Recall from the "skolemization" theorem that, for example,

$$\begin{aligned}
\{\Gamma, \exists x \phi(x)\} \text{ is inconsistent} \quad &\Leftrightarrow \quad \Gamma, \exists x \phi(x) \models \exists y(y \neq y) \\
&\Leftrightarrow \quad \Gamma, \phi(c) \models \exists y(y \neq y) \\
&\Leftrightarrow \quad \{\Gamma, \phi(c)\} \text{ is inconsistent,}
\end{aligned}$$

where c does not occur in Γ.

Preservation Theorems (cont.)

Math 260C - Mathematical Logic

April 12, 1989

Theorem: A theory T is preserved under unions of chains iff T has a set of Π_2 axioms.

Proof: $\Leftarrow$: already done
$\Rightarrow$: let Δ be the set of sentences logically equivalent to the set of Π_2-sentences. In particular, Δ is closed under finite disjunction.[1] Suppose that $\mathcal{A} \models T$ and that for all $\phi \in \Delta$, if $\mathcal{A} \models \phi$ then $\mathcal{B} \models \phi$. Then we want to show that $\mathcal{B} \models T$. Now, every Σ_2-sentence true in $\mathcal{B}$ is also true in $\mathcal{A}$.[2]

> *Claim:* There are structures $\mathcal{A}^{'}$ and $\mathcal{B}^{'}$ such that
>
> 1. $\mathcal{A} \equiv \mathcal{A}^{'}$,
> 2. $\mathcal{B} \prec \mathcal{B}^{'}$, and
> 3. $\mathcal{B} \subseteq \mathcal{A}^{'} \subseteq \mathcal{B}^{'}$.
>
> *Proof*: Let L be the language of T, and let L_B be $L \cup \{c_b : b \in |\mathcal{B}|\}$ where the c_b's are new constant symbols not in L. Let $T_1 = \mathrm{Th}(\mathcal{A})$ (a complete theory), and let T_2 be the set of Π_1-sentences true in $\mathcal{B}_B$ where $\mathcal{B}_B$ is the structure $\mathcal{B}$ extended to L_B.
>
> > *Sub-claim 1:* $T_1 \cup T_2$ is a consistent theory.
> >
> > *Proof*: Any finite subset of T_2 is of the form

[1] I.e., any finite disjunction of Π_2-sentences can be converted to a Π_2-sentence by prenex operations.

[2] Since every Π_2-sentence true in $\mathcal{A}$ is true in $\mathcal{B}$. (A Σ_2-sentence is of the form $\exists x_1 \ldots \exists x_n \forall y_1 \ldots \forall y_m \phi$ where ϕ is quantifier free.)

$$\{\psi_1(c_{b_1},\ldots,c_{b_n}),\ldots\psi_k(c_{b_1},\ldots,c_{b_n})\}.$$

(Note that these ψ's are universal.) This subset is consistent with T_1 iff (backwards skolemizing)

$$\exists x_1\ldots\exists x_n(\psi_1(\vec{x})\wedge\ldots\wedge\psi_k(\vec{x}))$$

is. (Note that by prenex operations, this is equivalent to a Σ_2-sentence in the language L.) Since

$$\mathcal{B}_B\models\psi_1(b_1,\ldots,b_n)\wedge\ldots\wedge\psi_k(b_1,\ldots,b_n),$$

we have

$$\mathcal{B}\models\exists x_1\ldots\exists x_n(\psi_1(\vec{x})\wedge\ldots\wedge\psi_k(\vec{x})).$$

Hence,

$$\mathcal{A}\models\exists x_1\ldots\exists x_n(\psi_1(\vec{x})\wedge\ldots\wedge\psi_k(\vec{x}))$$

since every Σ_2-sentence true in $\mathcal{B}$ is true in $\mathcal{A}$. This formula is already in T_1 (since it is in $\mathrm{Th}(\mathcal{A})$) and is certainly consistent with T_1. $\Box$

So let $\mathcal{A}'_B\models T_1\cup T_2$. Without loss of generality, (making an isomorphic copy if necessary) $c_b^{\mathcal{A}'_B}=b$ for all $b\in|\mathcal{B}|$. So, if $\mathcal{A}'$ is the restriction of $\mathcal{A}'_B$ to the language L, $\mathcal{B}\subseteq\mathcal{A}'$. Also, $\mathcal{A}'\equiv\mathcal{A}$ because $T_1=\mathrm{Th}(\mathcal{A})$.

Any Π_1-sentence true in $\mathcal{B}_B$ is also true in $\mathcal{A}'_B$ (since $\mathcal{A}'_B\models T_2$). So any existential sentence true in $\mathcal{A}'_B$ is also true in $\mathcal{B}_B$.

Now, further expand the language L_B to a new language L_A by adding new constant symbols c_a for each $a\in|\mathcal{A}'|\setminus|\mathcal{B}|$ (keeping the old c_b's).[3] Let $\mathcal{A}'_A$ be $\mathcal{B}'_A$ expanded to the language L_A. Let $D(\mathcal{A}'_A)$ be the set of quantifier free sentences true in $\mathcal{A}'_A$.[4] Let T' be $D(\mathcal{A}'_A)\cup\mathrm{Th}(\mathcal{B}_B)$.[5]

[3] At this point, we've dropped one level of quantifiers, and we'll proceed the same as before except with atomic formulas instead of Π_1-formulas.

[4] Essentially the same as the atomic diagram.

[5] To sharpen the analogy between this case and the previous case, we could say that $T_3=\mathrm{Th}(\mathcal{B}_B)$ and $T_4=D(\mathcal{A}'_A)$.

Sub-claim 2: $T^{'}$ is a consistent theory.

Proof: Any finite subset of $D(\mathcal{A}^{'}_{A})$ is of the form

$$\{\psi_1(c_{a_1},\ldots,c_{a_n}),\ldots\psi_k(c_{a_1},\ldots,c_{a_n})\}.$$

where the ψ's are quantifier free and $a_1,\ldots,a_n \in |\mathcal{A}^{'}|\setminus|\mathcal{B}|$. This subset is consistent with $\mathrm{Th}(\mathcal{B}_B)$ iff

$$\exists x_1\ldots\exists x_n(\psi_1(\vec{x})\wedge\ldots\wedge\psi_k(\vec{x}))$$

is. Since any existential sentence true in $\mathcal{A}^{'}_{B}$ is true in $\mathcal{B}_B$, the above sentence is in $\mathrm{Th}(\mathcal{B}_B)$. So it is certainly consistent with $\mathrm{Th}(\mathcal{B}_B)$. □

So let $\mathcal{B}^{'}_{A} \models T^{'}$. Without loss of generality, (making an isomorphic copy if necessary) $c_a^{\mathcal{B}^{'}_{A}} = a$ for $a \in |\mathcal{A}^{'}|$. Let $\mathcal{B}^{'}$ be the restriction of $\mathcal{B}^{'}_{A}$ to the language L. Then $\mathcal{A}^{'} \subseteq \mathcal{B}^{'}$ since $\mathcal{A}^{'}_{A} \models D(\mathcal{A}^{'}_{A})$, and $\mathcal{B} \prec \mathcal{B}^{'}$ since $\mathcal{B}^{'}_{A} \models \mathrm{Th}(\mathcal{B}_B)$. □

Now, iterate the above process to get the structures $\mathcal{A}_1,\mathcal{A}_2,\ldots$ and $\mathcal{B}_0,\mathcal{B}_1,\ldots$ such that

$$\mathcal{B}=\mathcal{B}_0\subseteq\mathcal{A}_1\subseteq\mathcal{B}_1\subseteq\mathcal{A}_2\subseteq\mathcal{B}_2\subseteq\ldots$$

with $\mathcal{B}_i \prec \mathcal{B}_{i+1}$ and $\mathcal{A}\equiv\mathcal{A}_i$ for all i.

Then, by the definition of union of chains, $\bigcup_i \mathcal{A}_i = \bigcup_i \mathcal{B}_i$. Also, $\mathcal{A}_i \models T$ for all i. Since T is preserved under unions of chains, $\bigcup_i \mathcal{A}_i \models T$. And by the elementary chain theorem, $\mathcal{B} \prec \bigcup_i \mathcal{B}_i$. So $\mathcal{B} \equiv \bigcup_i \mathcal{B}_i \models T$. □

Preservation Theorems (cont.)

Math 260C - Mathematical Logic

April 14, 1989

Definition: Let $\mathcal{A}$ and $\mathcal{B}$ be two structures with the same language, L. $\mathcal{A}$ pos $\mathcal{B}$ means that if ϕ is a positive sentence in the language L and $\mathcal{A} \models \phi$, then $\mathcal{B} \models \phi$.

Definition: An *embedding* of $\mathcal{A}$ into $\mathcal{B}$ is a homomorphism of $\mathcal{A}$ onto (not necessarily 1-1) a substructure of $\mathcal{B}$.

Theorem: A theory T is preserved under homomorphisms iff T has a set of positive axioms.[1]

Proof: $\Leftarrow$: It suffices to show that if ϕ is a positive sentence, $\mathcal{A} \models \phi$, and $\mathcal{B}$ is a homomorphic image of $\mathcal{A}$, then $\mathcal{B} \models \phi$. Let $f : \mathcal{A} \to \mathcal{B}$ be a homomorphism.

> *Claim:* If $\phi(x_1, \ldots, x_n)$ is a positive formula, and $a_1, \ldots, a_n \in |\mathcal{A}|$, then if $\mathcal{A} \models \phi(a_1, \ldots, a_n)$, then $\mathcal{B} \models \phi(f(a_1), \ldots, f(a_n))$.
>
> *Proof*: By induction of the complexity of ϕ.

[1] The intuitive reason behind restricting the axioms to being positive is that, by definition, a homomorphism preserves the truthness of atomic formulas; it doesn't necessarily preserve the truthness of negated atomic formulas or other non-positive formulas. For instance, a set of axioms for groups (with language $e, \cdot$) is

$\forall x(e \cdot x = x)$,
$\forall x(x \cdot e = x)$,
$\forall x \exists y(x \cdot y = e)$,
$\forall x \exists y(y \cdot x = e)$, and
$\forall x \forall y \forall z(x \cdot (y \cdot z) = (x \cdot y) \cdot z)$.

So groups are closed under homomorphisms.

Basis: ϕ is atomic and of the form $P(t_1(\vec{a}), \ldots, t_k(\vec{a}))$ (where P is possibly '='). Now,

$$f(t_i^{\mathcal{A}}(a_1, \ldots, a_n)) = t_i^{\mathcal{B}}(f(a_1), \ldots, f(a_n))$$

by induction on the complexity of t_i and using the fact that homomorphisms respect functions and constants. So if

$$\mathcal{A} \models P(t_1(\vec{a}), \ldots, t_k(\vec{a})),$$

then

$$\mathcal{B} \models P(f(t_1(\vec{a})), \ldots, f(t_k(\vec{a})))$$

by the definition of homomorphism. And so

$$\mathcal{B} \models P((t_1(f(a_1), \ldots, f(a_n))), \ldots, t_k(f(a_1), \ldots, f(a_n))).$$

Induction: The cases in which ϕ is $\psi \wedge \chi$ or is $\psi \vee \chi$ are easy by the induction hypothesis.

If ϕ is $\exists x \psi(x_1, \ldots, x_n, x)$, then

$$\begin{array}{lll}
\multicolumn{2}{l}{\mathcal{A} \models \phi(a_1, \ldots, a_n)} & \\
\quad \Leftrightarrow & \exists a \in |\mathcal{A}|,\ \mathcal{A} \models \psi(a_1, \ldots, a_k, a) & \text{by defn. of truth} \\
\quad \Rightarrow & \exists a \in |\mathcal{A}|,\ \mathcal{B} \models \psi(f(a_1), \ldots, f(a_k), f(a)) & \text{by ind. hyp.} \\
\quad \Rightarrow & \exists b \in |\mathcal{B}|,\ \mathcal{B} \models \psi(f(a_1), \ldots, f(a_k), b) & \text{take } b = f(a) \\
\quad \Leftrightarrow & \mathcal{B} \models \phi(f(a_1), \ldots, f(a_n)) & \text{by defn. of truth}
\end{array}$$

If ϕ is $\forall x \psi(x_1, \ldots, x_n, x)$, then

$$\begin{array}{lll}
\multicolumn{2}{l}{\mathcal{A} \models \phi(a_1, \ldots, a_n)} & \\
\quad \Leftrightarrow & \forall a \in |\mathcal{A}|,\ \mathcal{A} \models \psi(a_1, \ldots, a_k, a) & \text{by defn. of truth} \\
\quad \Rightarrow & \forall a \in |\mathcal{A}|,\ \mathcal{B} \models \psi(f(a_1), \ldots, f(a_k), f(a)) & \text{by ind. hyp.} \\
\quad \Leftrightarrow & \forall b \in |\mathcal{B}|,\ \mathcal{B} \models \psi(f(a_1), \ldots, f(a_k), b) & \text{since } f \text{ is onto} \\
\quad \Leftrightarrow & \mathcal{B} \models \phi(f(a_1), \ldots, f(a_n)) & \text{by defn. of truth } \Box
\end{array}$$

As an aside, we have

Theorem: Let ϕ be a positive sentence, and let $\mathcal{A}$ be the structure with $|\mathcal{A}| = \{a\}$, and $P^{\mathcal{A}} = \{a\}^k$ for all k-ary P. Then $\mathcal{A} \models \phi$.

Proof: By induction on the complexity of ϕ, show that if $\phi(x_1, \ldots, x_n)$ is a positive formula, then $\mathcal{A} \models \phi(a, \ldots, a)$. $\Box$

Corollary: If Γ is a set of positive sentences, then Γ is consistent.

$\Rightarrow$: (Let A be $|\mathcal{A}|$, and let B be $|\mathcal{B}|$). Let L_A be the language $L \cup \{c_a : a \in A\}$.

Claim: If $\mathcal{A}$ pos $\mathcal{B}$, then there is an elementary extension $\mathcal{B}' \succ \mathcal{B}$ and an embedding f of $\mathcal{A}$ into $\mathcal{B}'$ such that

$$(\mathcal{A}, a)_{a \in A} \text{ pos } (\mathcal{B}', f(a))_{a \in A}$$

where $(\mathcal{A}, a)_{a \in A}$ is the expansion of $\mathcal{A}$ to the language L_A, and $(\mathcal{B}', f(a))_{a \in A}$ is $\mathcal{B}'$ expanded to the language L_A, with $c_a^{(\mathcal{B}', f(a))_{a \in A}} = f(a)$; i.e.,

$$\begin{tikzcd}
\mathcal{A} \arrow[dr, "f"] & \\
\mathcal{B} \arrow[r, phantom, "\prec"] & \mathcal{B}'
\end{tikzcd}$$

with $(\mathcal{A}, a)_{a \in A}$ pos $(\mathcal{B}', f(a))_{a \in A}$.

Proof: Let $L_B = L \cup \{d_b : b \in B\}$. Let T_1 be the set of positive sentences true in $(\mathcal{A}, a)_{a \in A}$ in the language L_A, and let T_2 be the set of sentences true in $(\mathcal{B}, b)_{b \in B}$ in the language L_B. (Note that we have picked the theories T_1 and T_2 according to what we want to satisfy.)

(Proof continued next time.)

Preservation Theorems (cont.)

Math 260C - Mathematical Logic

April 17, 1989

Last time, we were in the middle of proving that T is preserved under homomorphisms iff it has a set of positive axioms. We proved the easy direction ($\Leftarrow$), and started the other direction.

$\Rightarrow$ (cont.):

> *Claim 1:* If $\mathcal{A}$ pos $\mathcal{B}$, then there is an elementary extension $\mathcal{B}' \succ \mathcal{B}$ and an embedding f of $\mathcal{A}$ into $\mathcal{B}'$ such that
>
> $$(\mathcal{A}, a)_{a \in A} \text{ pos } (\mathcal{B}', f(a))_{a \in A}.$$
>
> *Proof*: Let $L_A = L \cup \{c_a : a \in A\}$, and let $L_B = L \cup \{d_b : b \in B\}$. Let $T_1 = \{$positive sentences in L_A true in $(\mathcal{A}, a)_{a \in A}\}$, and let $T_2 = \{$sentences in L_B true in $(\mathcal{B}, b)_{b \in B}\}$.
>
> > *Sub-claim:* $T_1 \cup T_2$ is consistent.
> >
> > *Proof*: Any finite subset of T_1 is of the form
> >
> > $$\{\psi_1(a_1, \ldots, a_n), \ldots, \psi_k(a_1, \ldots, a_n)\}.$$
> >
> > This is consistent with T_2 iff
> >
> > $$\exists x_1 \ldots \exists x_n (\psi_1(\vec{x}) \wedge \ldots \wedge \psi_k(\vec{x})) \qquad (1)$$
> >
> > is. Now, the ψ's are positive by the definition of T_1, so (1) is a positive sentence true in $\mathcal{A}$. Hence, since $\mathcal{A}$ pos $\mathcal{B}$, (1) is true in $\mathcal{B}$ and thus in T_2. □

Since $T_1 \cup T_2$ is consistent, it has a model. So let

$$\mathcal{B}^* = (\mathcal{B}, a^{'}, b)_{a \in A, b \in B} \models T_1 \cup T_2,$$

where, without loss of generality, $d_b^{\mathcal{B}^*} = b$. Then $\mathcal{B}^{'}$ is the restriction of $\mathcal{B}^*$ to the language L.[1] Now, define $f(a) = a^{'}$, the embedding of $(\mathcal{A}, a)_{a \in A}$ into $(\mathcal{B}, a^{'}, b)_{a \in A, b \in B}$. (Note that f is not necessarily onto.)

> *Sub-claim:* $(\mathcal{A}, a)_{a \in A}$ pos $(\mathcal{B}^{'}, f(a))_{a \in A}$ (in the language L_A).
>
> *Proof*: $a_1 = a_2 \Rightarrow f(a_1) = f(a_2)$ since f is a function. For any function symbol $g \in L$, and $a \in A$, $g^{\mathcal{A}}(a) = a_2$ for some $a_2 \in A$. So $a_2 = g(a)$ is in T_1, and so $\mathcal{B}^* \models a_2 = g(a)$. Similarly, $P(a) \in T \Rightarrow \mathcal{B}^* \models P(a)$ for P a positive formula involving a.
>
> This proves the sub-claim and claim 1. □

Claim 2: (Almost dual of claim 1.) If $\mathcal{A}$ pos $\mathcal{B}$, then there is an elementary extension $\mathcal{A}^{'} \succ \mathcal{A}$ and a mapping $g : B \to A^{'}$ such that

$$(\mathcal{A}, g(b))_{b \in B} \text{ pos } (\mathcal{B}, b)_{b \in B}.$$

Proof: Let $T_3 = \{\text{sentences in } L_A \text{ true in } (\mathcal{A}, a)_{a \in A}\}$, and let $T_4 = \{\neg\phi : \phi \text{ is positive in } L_B \text{ and } (\mathcal{B}, b)_{b \in B} \models \neg\phi\}$.

> *Sub-claim:* $T_3 \cup T_4$ is consistent.
>
> *Proof*: Any finite subset of T_4 is of the form
>
> $$\{\neg\psi_1(d_{b_1}, \ldots, d_{b_n}), \ldots, \neg\psi_k(d_{b_1}, \ldots, d_{b_n})\}$$
>
> where the ψ's are positive. But,

[1] Note that $c_a^{\mathcal{B}^*} = a^{'}$ may not equal a. The reason is that T_1 contains only positive sentences, while T_2 doesn't have this restriction. I.e., it is possible for $a_1^{'} = a_2^{'}$ even though $\mathcal{A} \models a_1 \neq a_2$. But this is not possible for the b's.

$$\begin{aligned}\mathcal{B} &\models \exists x_1 \ldots \exists x_n(\neg\psi_1(\vec{x}) \wedge \ldots \wedge \neg\psi_k(\vec{x})) \\ &\models \neg\forall x_1 \ldots \forall x_n(\psi_1(\vec{x}) \vee \ldots \vee \psi_k(\vec{x})).\end{aligned}$$

Now any positive sentence false in $\mathcal{B}$ is false in $\mathcal{A}$. So

$$\mathcal{A} \models \exists x_1 \ldots \exists x_n(\neg\psi_1(\vec{x}) \wedge \ldots \wedge \neg\psi_k(\vec{x})) \qquad (2)$$

because so does $\mathcal{B}$ and because the formula in (2) is logically equivalent to the negation of a positive sentence. So

$$\exists x_1 \ldots \exists x_n(\neg\psi_1(\vec{x}) \wedge \ldots \wedge \neg\psi_k(\vec{x}))$$

is in T_3 and hence consistent with T_3. And so, the finite subset of T_4 is consistent with T_3. □

Since $T_3 \cup T_4$ is consistent, it has a model. So let

$$\mathcal{A}^* = (\mathcal{A}', a, b')_{a\in A, b\in B} \models T_3 \cup T_4,$$

with $c_a^{\mathcal{A}^*} = a$, and $d_b^{\mathcal{A}^*} = b'$.[2] Let $g(b) = b'$, and let $\mathcal{A}'$ be the restriction of $\mathcal{A}^*$ to the language L. Then

$$(\mathcal{A}', g(b))_{b\in B} \text{ pos } (\mathcal{B}, b)_{b\in B}$$

since if any positive sentence ϕ in L_B is false in $(\mathcal{B}, b)_{b\in B}$, then $\neg\phi$ is in T_4, and hence ϕ is false in $(\mathcal{A}', g(b))_{b\in B}$. □

(Proof of ⇒ part of theorem continued.) Let $\mathcal{A} \models T$, and $\mathcal{A}$ pos $\mathcal{B}$. By the lemma (of a few lectures ago), it suffices to show that $\mathcal{B} \models T$. By iterating the two claims above, build two elementary chains of structures:

[2] Note that $\mathcal{B} \models b_1 \neq b_2$ means that $d_{b_1} \neq d_{b_2}$ is in T_4. So $\mathcal{A}^* \models b_1' \neq b_2'$. Note also, that we could have $\mathcal{B} \models b_1 = h(b_2)$ for $h \in L$, but $\mathcal{B}^* \models b_1' \neq h(b_2')$. I.e., things that weren't equal in $\mathcal{B}$, aren't equal in $\mathcal{A}^*$. But things that were equal in $\mathcal{B}$, could become unequal in $\mathcal{A}^*$.

$$\begin{tikzcd}
\mathcal{A}=\mathcal{A}_0 \arrow[drr, "f_0"] & \prec & \mathcal{A}_1 \arrow[drr, "f_0"] & \prec & \mathcal{A}_2 & \cdots \\
\mathcal{B}=\mathcal{B}_0 & \prec & \mathcal{B}_1 \arrow[u, "g_1"'] & \prec & \mathcal{B}_2 \arrow[u, "g_1"'] & \cdots
\end{tikzcd}$$

with

$$\begin{array}{rcl}
(\mathcal{A}_0, a)_{a\in A} & \text{pos} & (\mathcal{B}_1, f_0(a))_{a\in A} \\
(\mathcal{A}_1, a, g(b^{'}))_{a\in A_0, b'\in B_1} & \text{pos} & (\mathcal{B}_1, f_0(a), b^{'})_{a\in A, b'\in B_1} \\
(\mathcal{A}_1, a, g(b^{'}), a^{'})_{a\in A_0, b'\in B_1, a'\in A_1} & \text{pos} & (\mathcal{B}_2, f_0(a), b^{'}, f_1(a^{'}))_{a\in A, b'\in B_1, a'\in A_1} \\
 & \vdots &
\end{array}$$

Let $\mathcal{A}_\omega = \bigcup_i \mathcal{A}_i$. By the elementary chain theorem, $\mathcal{A}_\omega \equiv \mathcal{A}$; hence $\mathcal{A}_\omega \models T$. Similarly, let $\mathcal{B}_\omega = \bigcup_i \mathcal{B}_i$. Again, by the elementary chain theorem, $\mathcal{B}_\omega \equiv \mathcal{B}$. In order to show that $\mathcal{B}_\omega \models T$, we need to show that $f_\omega = \bigcup_i f_i$ is a homomorphism of $\mathcal{A}_\omega$ onto $\mathcal{B}_\omega$.

> *Claim 1:* f_n extends f_{n-1}; i.e., they agree on their common domain.
>
> *Proof*: Suppose that $a^{'} \in A_1$, $a \in A_0$, and $a^{'} = a$. Then we need to show that $f_0(a) = f_1(a^{'}) = f_1(a)$. Now, $c_a = c_{a'}$ is a positive sentence in $(L_{A_0})_{A_1}$. So $\mathcal{A}_1 \models c_a = c_{a'}$, and so $\mathcal{B}_2 \models c_a = f_1(a^{'})$. Since $c_a^{\mathcal{B}_2} = f_0(a)$, $f_0(a) = f_1(a^{'})$. And since f is a function, $f_1(a^{'}) = f_1(a)$. $\Box$

(Proof continued next time. We still have to show that the f's are homomorphisms and that they're onto.)

Preservation Theorems (cont.)

Math 260C - Mathematical Logic

April 19, 1989

Last time, we were in the process of concluding the ⇒ part of the proof that T is preserved under homomorphisms iff it has a set of positive axioms. We built two elementary chains of structures:

$$\begin{tikzcd}
\mathcal{A} = \mathcal{A}_0 \arrow[drr, "f_0"] & \prec & \mathcal{A}_1 \arrow[drr, "f_0"] & \prec & \mathcal{A}_2 & \cdots \\
\mathcal{B} = \mathcal{B}_0 & \prec & \mathcal{B}_1 \arrow[u, "g_1"'] & \prec & \mathcal{B}_2 \arrow[u, "g_1"'] & \cdots
\end{tikzcd}$$

such that the f_i's are embeddings, the g_i's are function, and

$$\begin{array}{rcl}
\mathcal{A} & \text{pos} & \mathcal{B} \\
(\mathcal{A}_0, a)_{a \in A} & \text{pos} & (\mathcal{B}_1, f_0(a))_{a \in A} \\
(\mathcal{A}_1, a, g(b^{'}))_{a \in A_0, b^{'} \in B_1} & \text{pos} & (\mathcal{B}_1, f_0(a), b^{'})_{a \in A, b^{'} \in B_1} \\
(\mathcal{A}_1, a, g(b^{'}), a^{'})_{a \in A_0, b^{'} \in B_1, a^{'} \in A_1} & \text{pos} & (\mathcal{B}_2, f_0(a), b^{'}, f_1(a^{'}))_{a \in A, b^{'} \in B_1, a^{'} \in A_1} \\
 & \vdots &
\end{array}$$

We also showed that

$$\mathcal{A}_\omega = \bigcup_i \mathcal{A}_i \equiv \mathcal{A} \models T$$

and that

$$\mathcal{B}_\omega = \bigcup_i \mathcal{B}_i \equiv \mathcal{B}.$$

To continue with the proof, we'll reprove, in a better way, the following claim.

Claim 1: $f_n \supseteq f_{n-1}$.

Proof: (By example, for the case $n = 0$.) Let $x \in A_0 \subseteq A_1$. Let c^0_x be the constant symbol for x as a member of A_0, and let c^1_x be the constant symbol for x as a member of A_1.[1] Now,

$$(\mathcal{A}_1, a, g_1(b^{'}), a^{'})_{a \in A_0, b' \in B_1, a' \in A_1} \models c^0_x = x.^2$$

So

$$(\mathcal{B}_1, f_0(a), b^{'}, f_1(a^{'}))_{a \in A_0, b' \in B_1, a' \in A_1} \models c^0_x = x$$

by the fourth pos relationship above, where $y = f_1(x)$, and

$$(c^0_x)^{(\mathcal{B}_1, f_0(a), b^{'}, f_1(a^{'}))_{a \in A_0, b' \in B_1, a' \in A_1}} = f_0(x),$$

by the second pos relationship above. So $f_0(x) = f_1(x)$. □

Claim 2: $g_n^{-1} \subseteq f_n$.

Proof: (By example for the case $n = 1$.) Suppose that $g_1(y) = x$. Then we want to show that $f_1(x) = y$. Let $y \in B_1$, and $x \in A_1$.[3] Let d^1_y be the constant symbol for y as an element of B_1. Then

$$(A_1, a, g_1(b^{'}))_{a \in A_0, b' \in B_1} \models d^{'}_y = x$$

since

$$(d^{'}_y)^{(A_1, a, g_1(b^{'}))_{a \in A_0, b' \in B_1}} = g(y) = x.$$

So

$$(B_1, f_0(a), b^{'})_{a \in A_0, b' \in B_1} \models d^{'}_y = z$$

[1] We don't actually need c^1_x, but it clarifies the point that there really are two different constant symbols for x in the two models $\mathcal{A}_0$ and $\mathcal{A}_1$.

[2] Remember that $c^0_x = x$ for x an element of the universe is shorthand for $c^0_x = x_1\ s[x/x_1]$.

[3] Note that x may or may not be in A_0.

by the third pos relationship above, where $z = f_1(x)$. Now,

$$(d_y')^{(B_1,f_0(a),b')_{a\in A_0,b'\in B_1}} = y,$$

and $y = z = f_1(x)$. So $g_1^{-1} \subseteq f_1$. □

So f_ω is a function by claim 1. And f_ω is onto by claim 2 because the range of f_n includes the domain of $g_n = B_n$.

Claim 3: f_ω is a homomorphism.

Proof: f_ω is onto. Also, any positive sentence with elements from $|\mathcal{B}_\omega|$ uses only elements from B_n for large enough n. Now use the fact that f_n is an embedding; i.e., it preserves positive sentences. □

To conclude, we have that $\mathcal{A} \models T$ by hypothesis. $\mathcal{A}_\omega \models T$ since $\mathcal{A}_\omega \equiv \mathcal{A}$. And $\mathcal{B}_\omega \models T$ since $f_\omega : \mathcal{A}_\omega \overset{\text{homo.}}{\to} \mathcal{B}_\omega$ and since T is preserved under homomorphisms by hypothesis. So $\mathcal{B} \models T$ since $\mathcal{B} \equiv \mathcal{B}_\omega$. Hence by the lemma, T has a set of positive axioms. □

Gödel's Incompleteness Theorems

Math 260C - Mathematical Logic

April 19, 1989

In discussing incompleteness, we will work in the theory Q. Q has the language $0, S, +, \cdot$ and axioms:

$$\begin{array}{l} \forall x \forall y (Sx = Sy \to x = y) \\ \forall x (0 \neq Sx) \\ \forall x (x \neq 0 \to \exists y (x = Sy)) \\ \forall x (x + 0 = x) \\ \forall x \forall y (x + Sy = S(x + y)) \\ \forall x (x \cdot 0 = 0) \\ \forall x \forall y (x \cdot Sy = x \cdot y + y) \end{array}$$

Q is weak in the sense that it can't prove things such as addition being commutative.[1] Later, we'll see that in another sense, Q is strong. It provides a nice inductive axiomatization of $+$ and $\cdot$ along with some induction axioms.

A crucial element of the language is $\cdot$. Without it, we have models which are well understood:

- $\mathrm{Th}(N, 0, S, \leq)$ is decidable and admits elimination of quantifiers.
- $\mathrm{Th}(N, 0, S, +, \leq)$ is decidable and model-complete.
- $\mathrm{Th}(N, 0, S, +, \leq, \equiv_p)_{p \text{ prime}}$[2] admits elimination of quantifiers.

But with $\cdot$ in the language, things become undecidable. Part of the reason for this is that we can code recursive functions.

Definition: Let $f : N^k \to N$. Q (or in general any theory) can *represent* f iff there is a formula $\phi(x_1, \ldots, x_k, y)$ such that for every $n_1, \ldots, n_k \in N$, if $m = f(n_1, \ldots, n_k)$, then

[1] This can be shown by building non-standard models.

[2] $x \equiv_p y$ means $x \equiv y$ mod p.

$$Q \vdash \forall x(\phi(S^{n_1}0, \ldots, S^{n_k}0, x) \leftrightarrow x = S^m 0).$$

Or, in other words,

$$\begin{gathered} Q \vdash \phi(S^{n_1}0, \ldots, S^{n_k}0, S^m 0), \text{ and} \\ Q \vdash \forall x(S^{n_1}0, \ldots, S^{n_k}0, x) \rightarrow x = S^m 0). \end{gathered}$$

The idea behind this definition is that ϕ codes the graph of f.[3]

We'll start our study of Q by looking for functions representable in Q. These will turn out to be exactly the recursive functions.

[3] $S^n 0$ is a term in the language of Q. We can't use n itself because it is not in the language.

Representability

Math 260C - Mathematical Logic

April 21, 1989

Definition: f is *representable* in Q iff there is a formula $\phi(\vec{x}, y)$ such that

1. $\phi(\vec{x}, y)$ defines the graph of $f(\vec{x}) = y$ (in the standard model), and
2. for all $n_1, \ldots, n_k \in N$ if $m = f(\vec{n})$, then

$$Q \vdash \forall y(\phi(S^{n_1}0, \ldots, S^{n_k}0, y) \leftrightarrow y = S^m 0).$$

In fact, (2) implies (1) since Q is a true theory.[1]

Definition: f is *strongly representable* in Q iff there is a formula $\phi(\vec{x}, y)$ such that (1) and (2) as above hold and

3. $Q \vdash \forall \vec{x} \exists ! y \phi(x_1, \ldots, x_k, y)$.

(Recall that $\exists ! y \phi$ is shorthand for $\exists y \phi \wedge \forall y \forall y' (\phi \wedge \phi(y'/y) \rightarrow y = y')$.)

Definition: f is *definable* in Q iff there is a formula $\phi(\vec{x}, y)$ such that (1) and (3) as above hold.

Theorem: If f is representable, then f is strongly representable (but perhaps with a different ϕ).[2]

Proof: Let f be represented by ϕ. Let $\psi(\vec{x}, y)$ be

$$(\exists ! y \phi(\vec{x}, y) \rightarrow \phi(\vec{x}, y)) \wedge (\neg \exists ! y \phi(\vec{x}, y) \rightarrow y = 0).$$

Then

[1] Q being a *true* theory means that the axioms of Q are true in the standard model. Also, note that ϕ doesn't say anything about non-standard elements. The term $S^{n_i}0$ for $n_i \in N$ denotes the integer n_i. All we can say about "standardness" is that ϕ applied to standard arguments produces a standard result.

[2] Note that the converse is obvious.

1. ψ defines the graph of f because ϕ does.
2. If $n_1, \ldots, n_k \in N$ and $m = f(n_1, \ldots, n_k)$ then

$$Q \vdash \forall y(\phi(S^{n_1}0, \ldots, S^{n_k}0, y) \leftrightarrow y = S^m 0),$$

and so

$$Q \vdash \forall y(\psi(S^{n_1}0, \ldots, S^{n_k}0, y) \leftrightarrow y = S^m 0).$$

3. If there is a unique y such that $\phi(\vec{x}, y)$, then y is defined by $\phi(\vec{x}, y)$. If not, then y is 0. Either way, y is unique; i.e.,

$$Q \vdash \exists ! y \psi(\vec{x}, y). \ \square$$

Theorem: All representable functions are definable, but not vice versa.
Proof: Later.

The key difference between representable and definable functions is that for particular arguments $n_1, \ldots, n_k$ to representable functions, Q can prove that m is the unique result.

Theorem: If f is representable, then f is recursive.
Proof: Given $n_1, \ldots, n_k$, we find $f(n_1, \ldots, n_k)$ by enumerating all theorems[3] of Q until we find a theorem of the form

$$\forall y(\phi(S^{n_1}0, \ldots, S^{n_k}0, y) \leftrightarrow y = S^m 0).$$

Then $m = f(n_1, \ldots, n_k)$. $\square$

Our goal now is to show that every recursive function is representable. Recall that the recursive functions are precisely the base functions $S, +, \cdot, \dot{-}$, and $\Pi^m_k(x_1, \ldots, x_m) = x_k$ closed under minimization and composition. So we have to show that these notions are representable in Q.

1 Proposition: $S(x) = x + 1$ is representable in Q.
Proof: Let $\phi(x, y)$ be $Sx = y$. Then

[3] Recall a result we showed last quarter: If Γ is a recursive set of sentences, then the set of logical consequences (theorems) of Γ is r.e.

1. For all n,

$$Q \vdash S(S^n0) = S^{n+1}0$$

   since $S(S^n0)$ and $S^{n+1}0$ are identical terms.

2. For all n,

$$Q \vdash S(S^n0) = y \rightarrow y = S^{n+1}0$$

   by the equality axioms. □

2 Proposition: $+$ is representable in Q.
Proof: Let $\phi(x, y, z)$ be $x + y = z$.

Claim: If $n, m \in N$, then

$$Q \vdash S^n0 + S^m0 = S^{n+m}0.$$

Proof: By induction on m.

Basis: $m = 0$. Then

$$Q \vdash S^n0 + 0 = S^{n+1}0$$

by the axiom $\forall x(x + 0 = x)$.

Induction:

$$\begin{align} Q &\vdash S^n0 + S^{m+1} = S(S^n0 + S^m0) \tag{1} \\ Q &\vdash S^n0 + S^{m+1} = S(S^{n+m}0) \tag{2} \\ Q &\vdash S^n0 + S^{m+1} = S^{n+m+1}0 \tag{3} \end{align}$$

(1) is by the axiom $\forall x \forall y(x + Sy = S(x + y))$. (2) is by induction. And (3) is since $S(S^{n+m}0)$ and $S^{n+m+1}0$ are identical terms. □

The claim establishes the existence of ϕ. So we still have to show the uniqueness of ϕ. Now,

$$Q \vdash S^n 0 + S^m 0 = S^{n+m} 0$$

for all $n, m \in N$; hence

$$Q \vdash \forall z (S^n 0 + S^m 0 = z \rightarrow z = S^{n+m} 0)$$

since

$$Q \vdash \forall u \forall v \forall x \forall x^{'} (u + v = x \wedge u + v = x^{'} \rightarrow x = x^{'})$$

by the equality axioms. □

3 Proposition: $\cdot$ is representable in Q.
Proof: Let $\phi(x, y, z)$ be $x \cdot y = z$.

> *Claim:* If $n, m \in N$, then
>
> $$Q \vdash S^n 0 \cdot S^m 0 = S^{nm} 0.$$
>
> *Proof*: By induction on m.
>
> *Basis*: $m = 0$. Then
>
> $$Q \vdash S^n 0 \cdot 0 = 0$$
>
> by the axiom $\forall x (x \cdot 0 = 0)$.
>
> *Induction*:
>
> $$\begin{array}{lll} Q & \vdash & S^n 0 \cdot S^{m+1} = (S^n 0 \cdot S^m 0) + S^n 0 \quad \text{by } \forall x \forall y (x \cdot Sy = x \cdot y + x) \\ Q & \vdash & S^n 0 \cdot S^{m+1} = S^{nm} 0 + S^n 0 \quad \text{by induction} \\ Q & \vdash & S^n 0 \cdot S^{m+1} = S^{nm+n} 0 \quad \text{by proposition 2. } \square \end{array}$$

So

$$Q \vdash S^n 0 \cdot S^m 0 = S^{nm} 0$$

for all $n, m \in N$; hence

$$Q \vdash \forall x(S^n 0 \cdot S^m 0 = x \to x = S^{nm} 0)$$

since

$$Q \vdash \forall u \forall v \forall x \forall x^{'}(u \cdot v = x \wedge u \cdot v = x^{'} \to x = x^{'})$$

by the equality axioms. □

4 Proposition: I_k^m is representable in Q.
Proof: Let $\phi(x_1, \ldots, x_m, y)$ be $y = x_k$. Then

$$Q \vdash \forall y(y = S^n 0 \leftrightarrow y = S^n 0)$$

by the equality axioms. □

Representability (cont.)

Math 260C - Mathematical Logic

April 24, 1989

Last time we showed that all of the base functions for the class of recursive functions except $\dot{-}$ were representable in Q. $\dot{-}$ is harder and we'll leave it for later. Now we confront composition and minimization.

5 Proposition: If $i \neq j$, then $Q \vdash S^i 0 \neq S^j 0$.

Proof: Without loss of generality, let $i < j$. Use induction on i.

Basis: $i = 0$. We need to show that $Q \vdash 0 \neq S^j 0$. We have an axiom $\forall x(Sx \neq 0)$ which does this. $S^j 0$ is $S(S^{j-1}0)$ since $j > 0$.

Induction: $i > 0$.

$$Q \vdash S^i 0 = S^j 0 \rightarrow S^{i-1}0 = S^{j-1}0$$

from an axiom. And by the induction hypothesis (which says $Q \vdash S^{i-1}0 \neq S^{j-1}0$), $Q \vdash S^i 0 \neq S^j 0$. □

Theorem: If $g : N^k \rightarrow N$, and $h_1, \ldots, h_k : N^p \rightarrow N$, and $g, h_1, \ldots, h_k$ are all representable, then

$$f(\vec{x}) = g(h_1(\vec{x}), \ldots, h_k(\vec{x}))$$

is also representable. (I.e., representable functions are closed under composition.)

Proof: Suppose that $B_i(x_1, \ldots, x_p, y_i)$ represents h_i and that $A(y_1, \ldots, y_k, z)$ represents g. Then let $C(x_1, \ldots, x_p, z)$ be[1]

[1] We could also use $C(x_1, \ldots, x_p, z) =$

$$\forall y_1 \ldots \forall y_k (\bigwedge_{i=1}^{k} B_i(x_1, \ldots, x_p, y_i) \rightarrow A(y_1, \ldots, y_k, z)).$$

$$\exists y_1 \ldots \exists y_k (A(y_1,\ldots,y_k,z) \wedge \bigwedge_{i=1}^{k} B_i(x_1,\ldots,x_p,y_i)).$$

If $n_1,\ldots,n_p \in N$, $m_i = h_i(\vec{n})$ for $i = 1,\ldots,k$, and $q = g(\vec{m})$, then $q = f(\vec{n})$. Also, $Q \vdash C(S^{n_1}0,\ldots,S^{n_p},S^q0)$ since

$$Q \vdash A(S^{m_1}0,\ldots,S^{m_k}0,S^q0) \wedge \bigwedge_{i=1}^{k} B_i(S^{n_1}0,\ldots,S^{n_p}0,S^{m_i}0)$$

by the representability of the h_i's and g.

> *Convention:* Let $\underline{m}$ denote the same term as S^m0. So, for example, the above expression would be
>
> $$Q \vdash A(\underline{m_1},\ldots,\underline{m_k},\underline{q}) \wedge \bigwedge_{i=1}^{k} B_i(\underline{n_1},\ldots,\underline{n_p},\underline{m_i}).$$
>
> Note that $\underline{0}$ means 0.

Now we need to show uniqueness.

> *Claim:* $Q \vdash \forall z(C(\underline{n_1},\ldots,\underline{n_p},z) \to z = \underline{q})$.
>
> *Proof*:

$$\begin{array}{lll}
Q & \vdash & B_i(\underline{n_1},\ldots,\underline{n_p},y_i) \to y_i = \underline{m_i} \quad (1)\\
 & \vdash & A(y_1,\ldots,y_k,z) \wedge \bigwedge_{i=1}^{k} B_i(\underline{n_1},\ldots,\underline{n_p},y_i) \to \bigwedge_{i=1}^{k} y_i = \underline{m_i} \quad (2)\\
 & \vdash & A(y_1,\ldots,y_k,z) \wedge \bigwedge_{i=1}^{k} B_i(\underline{n_1},\ldots,\underline{n_p},y_i) \to A(\underline{m_1},\ldots,\underline{m_k},z) \quad (3)\\
 & \vdash & A(y_1,\ldots,y_k,z) \wedge \bigwedge_{i=1}^{k} B_i(\underline{n_1},\ldots,\underline{n_p},y_i) \to z = \underline{q} \quad (4)\\
 & \vdash & \exists y_1,\ldots \exists y_k(A(y_1,\ldots,y_k,z) \wedge \bigwedge_{i=1}^{k} B_i(\underline{n_1},\ldots,\underline{n_p},y_i)) \to z = \underline{q} \quad (5)
\end{array}$$

> (1) is because the h_i's are representable. (2) is just the conjunction of the formulas of the form of (1) with $A(y_1,\ldots,y_k,z)$ thrown in. (3) is by equality axioms. (4) is since A is the unique representation of g. And (5) since the y_i's are free on the right side of the →.[2] This completes the proof of the claim and the theorem. □

[2] The form of reasoning is that $Q \vdash C_m(\vec{y}) \to z = \underline{q}$ means that $Q \vdash (\forall \vec{y} C_m(\vec{y}) \to z = \underline{q})$. So by prenex operations and since the y's don't occur on the right side of the →, $Q \vdash \exists \vec{y} C_m(\vec{y}) \to z = \underline{q}$. This form of reasoning is called "∃-elimination".

Now we want to show that minimization is representable in Q. More precisely, we want to show that if $f(\vec{x}, y)$ is representable and is regular,[3] then $g(\vec{x}) = \mu y(f(\vec{x}, y) = 0)$ is representable. In doing this, we'll take a representation of f, say $A(\vec{x}, y, z)$, and create $B(\vec{x}, y)$ which represents g. Finding the y in the minimization is not hard; but showing that it is minimal and unique is. Before we do this, we'll have to build up some propositions and lemmas.

Definition: A relation $R \subseteq N^k$ is representable iff χ_R is.

6 Proposition: A k-ary relation R is representable iff there is a formula $\phi(x_1, \ldots, x_k)$ such that

1. ϕ defines[4] R, and
2. for all $n_1, \ldots, n_k$, either $Q \vdash \phi(\underline{n_1}, \ldots, \underline{n_k})$ or $Q \vdash \neg\phi(\underline{n_1}, \ldots, \underline{n_k})$.[5]

Proof: $\Rightarrow$: Suppose that $\psi(\vec{x}, y)$ represents χ_R. Then let $\phi(\vec{x})$ be $\psi(\vec{x}, \underline{1})$.
$\Leftarrow$: Given a ϕ such that (1) and (2) hold, let $\psi(\vec{x}, y)$ be

$$(y = 0 \wedge \neg\phi(\vec{x})) \wedge (y = \underline{1} \wedge \phi(\vec{x})). \ \Box$$

7 Proposition: The set of representable relations is closed under complementation, union, and intersection.
Proof: Trivial from proposition 6.[6]

Definition: $a < b$ is an abbreviation for $\exists x(Sx + a = b)$.

8 Proposition: If $i < j$, then $Q \vdash \underline{i} < \underline{j}$.
Proof: If $i < j$, then $m + 1 + i = j$ for some m. So $Q \vdash \underline{m + 1} + \underline{i} = \underline{j}$ since $+$ is representable in Q. Hence $Q \vdash S\underline{m} + \underline{i} = \underline{j}$, and $Q \vdash \exists x(Sx + \underline{i} = \underline{j})$. $\Box$

[3] I.e., for all $\vec{x}$, a y exists such that $f(\vec{x}, y) = 0$.

[4] ϕ defines R means that for all $\vec{n} \in N$, $N \models \phi(\vec{n}) \Leftrightarrow R(\vec{n})$.

[5] Note that $Q \vdash \forall\vec{x}(\phi(\vec{x} \vee \neg\phi(\vec{x}))$ does not imply that either $Q \vdash \phi(\vec{x})$ or $Q \vdash \neg\phi(\vec{x})$. If ϕ defines a relation R which is not recursive, then χ_R is definable in Q but not representable in Q. For example, if R expresses the relation that the n^{th} Turing machine halts on input a blank tape.

[6] If ϕ represents R, then $\neg\phi$ represents $N^k \setminus R$. If ϕ and ψ represent R and S, then $\phi \vee \psi$ represents $R \cup S$. Similary for $\cap$.

Lemma: For $i \geq 0$, $Q \vdash \forall x(Sx + \underline{i} = x + \underline{i+1})$.

Proof:

$$\begin{array}{rcl} Q \vdash Sx + S^i 0 & = & S(Sx + S^{i-1}0) \\ & \vdots & \\ & = & S^i(Sx + 0) \\ & = & S^{i+1}x \\ & = & S^{i+1}(x + 0) \\ & = & S^i(x + S0) \\ & \vdots & \\ & = & x + S^{i+1}0. \ \Box \end{array}$$

9 Proposition: For $i > 0$, $Q \vdash \forall x(x < \underline{i} \rightarrow x = \underline{0} \vee \ldots \vee x = \underline{i-1})$, and for $i = 0$, $Q \vdash \forall x(\neg x < \underline{0})$.

Proof: By induction on i.

Basis: $i = 0$. (We'll reason inside Q.) We have the axiom $x = 0 \vee \exists y(x = Sy)$.

Case 1. $x = 0$. Then $x < 0$ means that $\exists w(Sw + 0 = 0)$. Hence $\exists w(Sw = 0)$ by the axiom $\forall x(x + 0 = 0)$. But this contradicts the axiom $\forall x(Sx \neq 0)$. So $Q \vdash x = 0 \rightarrow \neg x < 0$.

Case 2. $\exists y(x = Sy)$. Then $x < 0$ means that $\exists w(Sw + x = 0) \Rightarrow \exists w(Sw + Sy = 0) \Rightarrow \exists w(S(Sw + y) = 0) \Rightarrow \exists v(Sv = 0) \Rightarrow$ contradiction. So $Q \vdash \exists y(x = Sy) \rightarrow \neg x < 0$.

This concludes the base case $i = 0$. Next lecture, we'll finish the proof.

Representability (cont.)

Math 260C - Mathematical Logic

April 26, 1989

9 Proposition: For $i = 0$, $Q \vdash \forall x(\neg x < \underline{0})$, and for $i > 0$, $Q \vdash \forall x(x < \underline{i} \rightarrow x = \underline{0} \vee \ldots \vee x = \underline{i-1})$.

Proof: By induction on i.

Basis: $i = 0$. We did this last time.

Induction: (Reasoning in Q again.) Suppose that $x < \underline{i+1}$. Then, since $Q \vdash x = 0 \vee \exists w(x = Sw)$ (this is actually an axiom), we have two cases.

Case 1. $x = 0$. Then there is nothing to prove.

Case 2. $x = Sw$ for some w. Now $x < \underline{i+1}$ means that $Sy + x = \underline{i+1}$ for some y. So $Sy + Sw = \underline{i+1}$. Hence $S(Sy + w) = \underline{i+1}$ by an addition axiom, and so $Sy + w = \underline{i}$ by the "S is 1-1" axiom. By the induction hypothesis, $w = \underline{0} \vee \ldots \vee w = \underline{i-1}$. So $Sw = \underline{1} \vee \ldots \vee Sw = \underline{i}$; i.e., $x = \underline{1} \vee \ldots \vee x = \underline{i}$. □

This proposition essentially says that there are no nonstandard numbers less than any standard number.

10 Proposition: $<$ is a representable relation.

Proof: Let ϕ be $x < y$. Suppose that $i, j \in N$.

Case 1. $i < j$. Then $Q \vdash \underline{i} < \underline{j}$ by proposition 8.

Case 2. $i \geq j$. Then for any $k = 0, \ldots, j-1$, $i \neq k$. Hence $Q \vdash \underline{i} \neq \underline{k}$ by proposition 5. Now,

$$Q \vdash x < \underline{j} \rightarrow x = \underline{0} \vee \ldots \vee x = \underline{j-1}.$$

So, $Q \vdash \neg \underline{i} < \underline{j}$.[1] $\square$

Theorem: Let $R(x_1, \ldots, x_k, x_{k+1})$ be a representable relation, and let

$$S(x_1, \ldots, x_k) \Leftrightarrow (\exists x_{k+1} < x_1) R(x_1, \ldots, x_{k+1}).$$

Then S is representable.[2]

Proof: Let $\phi(x_1, \ldots, x_{k+1})$ represent R, and let $\psi(x_1, \ldots, x_k)$ be $(\exists x_{k+1} < x_1)\phi(x_1, \ldots, x_{k+1})$. If $n_1, \ldots, n_k \in N$ and $S(\vec{n})$, then there is an $m < n_1$ such that $R(\vec{n}, m)$. So

$$Q \vdash \phi(\underline{n_1}, \ldots, \underline{n_k}, \underline{m}) \wedge \underline{m} < \underline{n_1}$$

by the representability of $<$. So $Q \vdash \psi(\underline{n_1}, \ldots, \underline{n_k})$. But if $\neg S(\vec{n})$, then

$$Q \vdash \neg\phi(\underline{n_1}, \ldots, \underline{n_k}, \underline{m})$$

for each $m = 0, \ldots, n_1 - 1$. So

$$Q \vdash x < \underline{n_1} \rightarrow x = \underline{0} \vee \ldots \vee x = \underline{n_1 - 1}.$$

Hence $Q \vdash \neg\psi(\underline{n_1}, \ldots, \underline{n_k})$. $\square$

Corollary: Representable relations are closed under bounded quantification.[3]

Lemma: For $i \geq 0$, $Q \vdash \forall x(\underline{i} < x \rightarrow \underline{i+1} = x \vee \underline{i+1} < x)$.

Proof: Fix i, and reason in Q. Suppose that $i < x$. Then $Sw + \underline{i} = x$ for some w. Now, by an axiom, $w = 0 \vee \exists y(w = Sy)$.

Case 1. $w = 0$. Then $S0 + \underline{i} = x$. But $S0 + \underline{i} = \underline{i+1}$, by the representability of $+$. So $x = \underline{i+1}$.

Case 2. $Sy = w$. Then $SSy + \underline{i} = x$. So $Sy + S\underline{i} = x$ by an earlier lemma (April 24). This is the same as $Sy + \underline{i+1} = x$ which implies by definition that $\underline{i+1} < x$. $\square$

[1] Note that we are using $<$ to represent three different things: (1) the relation "less than", (2) the name of the relation "less than", and (3) an abbreviation for $\exists x(Sx + \underline{a} = \underline{b})$.

[2] We can also define S in terms of R similarly with $\forall$ instead of $\exists$ since we can just represent $\forall$ as $\neg\exists\neg$.

[3] We can use any representable t to say $S(-) \Leftrightarrow (\exists x_{k+1} < t)R(-)$, and, since representable terms are closed under $+$, $\cdot$, composition, etc., the proof is basically the same.

This lemma essentially says that nothing fits in between i and $i+1$.

Theorem: (Trichotomy) For $i \geq 0$, $Q \vdash \forall x(x < \underline{i} \vee x = \underline{i} \vee \underline{i} < x)$.

Proof: By induction on i.

Basis: $i = 0$. (Reason in Q.) By an axiom, $x = 0 \vee \exists y(x = Sy)$. If $x = 0$, then there is nothing to prove. If $x = Sy$ for some y, then $Sy + \underline{0} = x$. So $\underline{0} < x$ by definition.

Induction: We want to show that

$$Q \vdash \forall x(x < \underline{i+1} \vee x = \underline{i+1} \vee \underline{i+1} < x).$$

By the induction hypothesis, we have that

$$Q \vdash \forall x(x < \underline{i} \vee x = \underline{i} \vee \underline{i} < x).$$

So there are three cases to consider.

Case 1. $x < \underline{i}$. By proposition 9,

$$x < \underline{i} \rightarrow x = \underline{0} \vee \ldots \vee x = \underline{i-1}.$$

And by the representability of $<$, we have

$$\underline{0} < \underline{i+1} \wedge \underline{1} < \underline{i+1} \wedge \ldots \wedge \underline{i-1} < \underline{i+1}.$$

So $x < \underline{i} \rightarrow x < \underline{i+1}$.

Case 2. $x = \underline{i}$. Then $x < \underline{i+1}$ by a similar argument.

Case 3. $\underline{i} < x$. Then either $x = \underline{i+1}$ or $\underline{i+1} < x$. In either case, there is nothing to prove. □

Representability (cont.)

Math 260C - Mathematical Logic

April 28, 1989

Theorem: Let f be a representable, regular function. Then $g(\vec{x}) = \mu y f(\vec{x}, y) = 0$ is representable.

Proof: Let $\phi(\vec{x}, y, z)$ represent f, and let $\psi(\vec{x}, y)$ be

$$\phi(\vec{x}, y, 0) \wedge (\forall z < y) \neg \phi(\vec{x}, z, 0).$$

> *Claim:* ψ represents g.
>
> *Proof*: If $m = g(\vec{n})$, then $f(\vec{n}, m) = 0$. So $Q \vdash \phi(\underline{\vec{n}}, \underline{m}, \underline{0})$ since ϕ is represented by f. Also, for each $m^{'} < n$, $Q \vdash \phi(\underline{\vec{n}}, \underline{m^{'}}, \underline{f(\vec{n}, m^{'})})$ since ϕ is represented by f. Hence $Q \vdash \neg\phi(\underline{\vec{x}}, \underline{m^{'}}, \underline{0})$ since $f(\vec{n}, m^{'}) \neq 0$. So $Q \vdash \psi(\underline{\vec{n}}, \underline{m})$.
>
> Now we have to show uniqueness; i.e., that
>
> $$Q \vdash \forall y(\psi(\underline{\vec{n}}, y) \to y = \underline{m}),$$
>
> or, in other words,
>
> $$Q \vdash \psi(\underline{\vec{n}}, y) \wedge \psi(\underline{\vec{n}}, \underline{m}) \to y = \underline{m}.$$
>
> We'll reason in Q. From the trichotomy theorem of last lecture, we have three cases to consider:
>
> **Case 1.** $y = \underline{m}$. Then there is nothing to prove.
>
> **Case 2.** $y > \underline{m}$. If $\psi(\underline{\vec{n}}, y)$ is true, then $(\forall z < y)\neg\phi(\underline{\vec{n}}, z, 0)$ holds, and in particular, $\neg\phi(\underline{\vec{n}}, \underline{m}, 0)$ holds. But this contradicts the fact that $Q \vdash \psi(\underline{\vec{n}}, \underline{m})$ implies that $\phi(\underline{\vec{n}}, \underline{m}, 0)$ holds. So $y \not> \underline{m}$.

Case 3. $y < \underline{m}$. Similar to case 2.

So $y = \underline{m}$. This completes the proof of the claim and the theorem. □

Corollary: Any recursive function is representable.

Homework: Show that $\dot{-}$ is representable.

Gödel's First Incompleteness Theorem

Math 260C - Mathematical Logic

April 28, 1989

We want to formalize in Q facts and definitions about provability in Q. In order to do so, we need to assign Gödel numbers to formulas, proofs, etc. For any string of symbols, α, we'll use the notation $\ulcorner \alpha \urcorner$ to denote the Gödel number of α. The following (from Boolos & Jeffrey) is one possible assignment of Gödel numbers to single symbols in the language:

$\ulcorner (\urcorner = 1$			
$\ulcorner) \urcorner = 2$	$\ulcorner , \urcorner = 29$		
$\ulcorner \wedge \urcorner = 3$	$\ulcorner \vee \urcorner = 39$	$\ulcorner \neg \urcorner = 399$	$\ulcorner \rightarrow \urcorner = 3999$
$\ulcorner \exists \urcorner = 4$	$\ulcorner \forall \urcorner = 49$		
$\ulcorner x_0 \urcorner = 5$	$\ulcorner x_1 \urcorner = 59$	$\ulcorner x_2 \urcorner = 599$	$\ldots$
$\ulcorner f_0^0 \urcorner = 6$	$\ulcorner f_1^0 \urcorner = 69$	$\ulcorner f_2^0 \urcorner = 699$	$\ldots$
$\ulcorner f_0^1 \urcorner = 68$	$\ulcorner f_1^1 \urcorner = 689$	$\ulcorner f_2^1 \urcorner = 6899$	$\ldots$
$\ulcorner f_0^2 \urcorner = 688$	$\ulcorner f_1^2 \urcorner = 6889$	$\ulcorner f_2^2 \urcorner = 68899$	$\ldots$
$\vdots$			
$\ulcorner A_0^1 \urcorner = 78$	$\ulcorner A_1^1 \urcorner = 789$	$\ulcorner A_2^1 \urcorner = 7899$	$\ldots$
$\ulcorner A_0^2 \urcorner = 788$	$\ulcorner A_1^2 \urcorner = 7889$	$\ulcorner A_2^2 \urcorner = 78899$	$\ldots$
$\vdots$			

The Gödel number of a string of symbols is formed by putting its constituent symbols' codes end to end (in base 10). For example, $\ulcorner \neg A_0^1(x_0) \urcorner = 39978152$.

Theorem: The following sets of integers are recursive, hence representable in Q:

1. $\{\ulcorner t \urcorner : t \text{ is a term}\}$,

2. $\{\ulcorner\alpha\urcorner : \alpha$ is a well formed formula$\}$, and

3. $\{\ulcorner\alpha\urcorner : \alpha$ is an axiom of $Q\}$.

Theorem: The following functions are recursive, hence representable in Q:

1. $\mathrm{Num}(n) = \ulcorner S^n 0\urcorner$,

2. $\mathrm{Sub}(\ulcorner A\urcorner, \ulcorner x\urcorner, \ulcorner t\urcorner) = \ulcorner A(t/x)\urcorner$, and

3. $\mathrm{Diag}(\ulcorner A\urcorner) = \mathrm{Sub}(\ulcorner A\urcorner, \ulcorner x_0\urcorner, \mathrm{Num}(\ulcorner A\urcorner))$.

Num is just the Gödel number of the term for the integer n; i.e., $\ulcorner \underline{n}\urcorner$. Sub is the Gödel number of the formula resulting from the substitution of term t for x in the formula A (with some convention on how to rename occurrences of variables in A in order to avoid clashes with variables in t). Diag is just $\ulcorner A(\underline{\ulcorner A\urcorner}/x_0)\urcorner$. It takes the Gödel number of a formula A and substitutes that number into the formula A itself for the variable x_0 and returns the Gödel number of the result. (Note that if A doesn't contain an occurrence of x_0 then $\mathrm{Diag}(\ulcorner A\urcorner) = \ulcorner A\urcorner$.)

Gödel's First Incompleteness Theorem (cont.)

Math 260C - Mathematical Logic

May 1, 1989

Recall last time, we introduced the functions Num : $N \to N$, Sub : $N \to N$, and Diag : $N^3 \to N$:

$$\begin{aligned}
\mathrm{Num}(n) &= \ulcorner S^n 0 \urcorner \\
\mathrm{Sub}(\ulcorner G \urcorner, \ulcorner x \urcorner, \ulcorner t \urcorner) &= \ulcorner G(t/x) \urcorner \\
\mathrm{Diag}(\ulcorner G \urcorner) &= \mathrm{Sub}(\ulcorner G \urcorner, \ulcorner x_0 \urcorner, \mathrm{Num}(\ulcorner G \urcorner))
\end{aligned}$$

Diagonalization Theorem: (Gödel) Let T be a theory whose language contains 0 and S, and suppose that Diag is representable in T. Let $B(y)$ be a formula whose only free variable is y. Then there is a sentence G such that $T \vdash G \leftrightarrow B(\underline{\ulcorner G \urcorner})$.[1]

Example: Consider the theory Q. Let $B(y)$ be $(\exists z < y)(z + z + 1 = y)$. So B represents the predicate "y is odd"; i.e., for any particular integer n, $B(n)$ is provable or disprovable.[2] The diagonalization theorem says that either

1. there is a G such that $Q \vdash G$ and $\ulcorner G \urcorner$ is odd, or
2. there is a G such that $Q \vdash \neg G$ and $\ulcorner G \urcorner$ is even.

This may be hard to understand, but consider the case where this is false. Then

1. every G such that $Q \vdash G$ would be such that $\ulcorner G \urcorner$ is even, and

[1] The theorem can be modified to allow B to have other free variables. In that case G would be a formula.

[2] Note that the diagonalization theorem itself doesn't say anything about B or G being provable or disprovable. But in the example, this just happens to be the case.

2. every G such that $Q \vdash \neg G$ would be such that $\ulcorner G \urcorner$ is odd.

This would be very surprising since then

$$\begin{aligned} \{\ulcorner G \urcorner : Q \vdash G\} &\subseteq \{\text{even integers}\}, \text{ and} \\ \{\ulcorner G \urcorner : Q \vdash \neg G\} &\subseteq \{\text{odd integers}\}. \end{aligned}$$

I.e., we wouldn't expect the last symbol of G to determine the provability of G. In fact we'll see that the proof of the diagonalization theorem does not use any fact about Gödel numbering except the fact that Diag is representable (i.e., a recursive function).

Proof: (of the diagonalization theorem) Let $A(x, y)$ represent Diag,[3] let $F(x_0)$ be the formula

$$\exists y(A(x_0, y) \wedge B(y)),$$

and let G be the sentence

$$F(\underline{\ulcorner F(x_0) \urcorner}/x_0).$$

> *Claim:* G is such that $T \vdash G \leftrightarrow B(\underline{\ulcorner G \urcorner})$.
>
> *Proof*: Let $m = \ulcorner F(x_0) \urcorner$, and let $k = \ulcorner G \urcorner$. So $k = \mathrm{Diag}(m)$, and so
>
> $$T \vdash \forall y(A(\underline{m}, y) \leftrightarrow y = \underline{k}). \tag{1}$$
>
> By the definition of F and G, G is
>
> $$\exists y(A(\underline{m}, y) \wedge B(y)).$$
>
> By (1),
>
> $$T \vdash G \leftrightarrow B(\underline{k}).$$
>
> I.e., $T \vdash G \leftrightarrow B(\underline{\ulcorner G \urcorner})$. The completes the proof of the claim and the theorem. □

[3] I.e., if $m = \mathrm{Diag}(n)$, then $T \vdash \forall y(A(\underline{n}, y) \leftrightarrow y = \underline{m})$.

The idea of the proof is that $F(x_0)$ is $B(S^{\mathrm{Diag}(x_0)}0)$. So $F(x_0) \approx B(\mathrm{Diag}(x_0))$ except that Diag is not a term in the language; so we "fake it" with A which represents F.[4] Then,

$$\begin{aligned} G &= F(\underline{\ulcorner F(x_0)\urcorner}/x_0) \\ &= B(\underline{\ulcorner F(\underline{\ulcorner F(x_0)\urcorner}/x_0)\urcorner}). \end{aligned}$$

G says "diagonalize F", or $\mathrm{Diag}(\ulcorner F\urcorner)$, or $\mathrm{Diag}(\ulcorner B(\mathrm{Diag}(x_0))\urcorner)$. And the result of this diagonalization is $B(\mathrm{Diag}(\ulcorner F\urcorner))$, or $B(\mathrm{Diag}(\ulcorner B(\mathrm{Diag}(x_0))\urcorner))$.

Theorem: Let T be a consistent theory containing Q. Then the set of Gödel numbers of theorems of T is not recursive (hence not representable in Q).

Proof: Suppose, for the sake of contradiction, that $C(x)$ represents the set of theorems of T. By the diagonalization theorem, there is a sentence G such that

$$Q \vdash G \leftrightarrow \neg C(\underline{\ulcorner G\urcorner}).$$

> *Claim:* $T \vdash G$.
>
> *Proof*: Assume otherwise. Then, since $T \nvdash G$ (i.e., G is not a theorem of T), $Q \vdash \neg C(\underline{\ulcorner G\urcorner})$ because C represents the set of theorems of T. So $Q \vdash G$ since $Q \vdash G \leftrightarrow \neg C(\underline{\ulcorner G\urcorner})$. Hence $T \vdash G$ since T contains Q. But this contradicts the assumption that $T \nvdash G$. □

So $T \vdash G$, hence $Q \vdash C(\underline{\ulcorner G\urcorner})$ since C represents the set of theorems of T. Thus $Q \vdash \neg G$, so $T \vdash \neg G$. So T is inconsistent. This contradicts our assumption that T is a consistent theory.[5] □

[4] If we had a function symbol in the language for Diag, we would have used it instead of A.

[5] That Q is consistent is not questioned since the natural numbers are a model of Q, and we believe in their consistency.

Gödel's First Incompleteness Theorem (cont.), Tarski's Theorem, ω-consistency

Math 260C - Mathematical Logic

May 5, 1989

Recall from last time,

Theorem: If T is a consistent extension of Q, then the set of Gödel numbers of theorems (of logical consequences) of T is not recursive.

Corollary: Q is not decidable. (The set of theorems of Q is not a recursive set.)

Corollary: $\mathrm{Th}(N, 0, S, +, \cdot)$ is not decidable. ("Arithmetic is undecidable". "Truth is not decidable".)
Proof: Let $T = \mathrm{Th}(N, 0, S, +, \cdot)$ and apply the theorem.[1]

Theorem: For any theory T, if T is complete, then T is decidable.
Proof: Two cases:

Case 1: T is consistent. Then the following algorithm "decides" T:

```
input: a sentence $A$

enumerate theorems of $T$ as $B_1, B_2, \ldots$ until either $A$ or $\neg A$ appears
if $A$ appears, output "yes"
else output "no"
```

Case 2: T is inconsistent. Then the following algorithm "decides" T:

[1] $\mathrm{Th}(N, 0, S, +, \cdot)$ is a consistent extension of Q in the sense that it contains all of the logical consequences of Q.

input: a sentence A

output "yes".

Note that in general, we can't tell whether T is consistent or not, so we don't know which case to use. But in either case, there *is* an algorithm. □

Gödel's First Incompleteness Theorem: (1931) There is no consistent, complete, axiomatizable theory extending Q.[2]

Proof: Suppose that T is such a theory. Then by the previous theorem, T is recursive. But this contradicts the first theorem. □

Note that Q is consistent and axiomatizable, but not complete, while the standard theory is consistent and complete, but not axiomatizable.[3]

Theorem: The following is undecidable: given a sentence A, is $Q \cup \{A\}$ consistent?

Proof: For all sentences B, $Q \vdash B$ iff $Q \cup \{\neg B\}$ is not consistent. But $\{B : Q \vdash B\}$ is undecidable. □

Theorem: In the language of Q, the set of valid formulas is undecidable.[4]

Proof: A decision procedure for the valid sentences in the language of Q yields a decision procedure for the theory Q (with axioms):

[2] "Axiomatizable" means "has an r.e. set of axioms". Note that if a theory has an r.e. set of axioms $\gamma_1, \gamma_2, \gamma_3, \ldots$, then a logically equivalent recursive set of axioms is $\gamma_1, \gamma_1 \wedge \gamma_2, \gamma_1 \wedge \gamma_2 \wedge \gamma_3, \ldots$. To see if a formula A is an axiom, just enumerate the γ_i's until the formula $\gamma_1 \wedge \gamma_2 \wedge \ldots \wedge \gamma_i$ has the same as or more symbols in it than A does.

[3] Before Gödel 's result, mathematicians wanted to find a set of axioms for a theory in which all the true statements about N were true. Gödel's incompleteness theorem results from work starting with Cantor in the late 1800's. Cantor used various non-constructive techniques (e.g. the axiom of choice) for proving things about set theory. This split mathematicians into two camps: one lead by Hilbert who adopted Cantor's methods, and the other, the constructivists, lead by Kronecker. Hilbert's program (ca. 1900) was to establish the consistency of Cantor's methods. In particular, he wanted to

- use constructive reasoning to establish the consistency of non-con-structive reasoning,
- establish the completeness of Peano arithmetic and set theory, and
- give algorithms for deciding the truth of sentences in the language of Peano arithmetic and the language of set theory.

[4] I.e., there are no axioms.

```
input: a formula $A$
output: "yes" if $Q \vdash A$, "no" if not

is $\bigwedge Q \to A$ valid?
if so, output "yes"
else output "no"
```

But this contradicts the fact that the set $\{A : \bigwedge Q \to A \text{ is valid}\}$ is undecidable.[5] □

Theorem: In the language $\{R\}$ where R is a binary relation symbol, the set of valid sentences is undecidable.

Proof: (Idea) Do the results of the last few lectures in a finitely axiomatizable fragment of set theory instead of Q. I.e., define $+$ and $\cdot$, etc.

Note that a theory T_R which has axioms to state that R is a dense linear ordering is decidable. The axioms say enough about R to make it decidable.

Tarski's Theorem: (on the Undefinability of Truth) In the language $0, S, +, \cdot$, there is no formula $T(x)$ such that for all sentences G, $N \models T(\underline{\ulcorner G \urcorner})$ iff $N \models G$ (where N is the standard model of the natural numbers).[6]

Proof: By Gödel's diagonalization theorem, (taking B to be $\neg T$) there is a G such that $Q \vdash G \leftrightarrow \neg T(\underline{\ulcorner G \urcorner})$. So $N \models G \leftrightarrow \neg T(\underline{\ulcorner G \urcorner})$. Which is a contradiction. □

Any r.e. predicate *is* definable in N. So truth is not r.e. We have shown that if $T \supseteq Q$ is consistent and axiomatizable, then T is not complete. Now we would like to find something that is independent of T.

Definition:

$$\mathrm{Prf}_T(x, y) = \{(x, y) : y = \ulcorner A \urcorner, \text{ and } x = \ulcorner \text{refutation of } T \cup \{\neg A\} \urcorner\}.$$

I.e., $\mathrm{Prf}_T(x, y)$ is a binary predicate denoting "x is a proof of y".

Now, $\mathrm{Prf}_T(x, y)$ is recursive, hence it is representable in Q by some formula, say $Prf_T(-, -)$. By Gödel's diagonalization theorem there is a formula G_T such that

[5] $\bigwedge Q$ is the conjunction of the axioms of Q.

[6] Note that this is stronger than saying that N is not decidable. There are things which are definable but not decidable. This is similar to the fact that there are things which are definable but not representable.

$$Q \vdash G_T \leftrightarrow \neg \exists x Prf_T(x, \underline{\ulcorner G_T \urcorner}).$$

Claim: $T \nvdash G_T$.

Proof: Suppose that $T \vdash G_T$. Then there is a refutation R of $T \cup \{\neg G_T\}$. So,

$$\begin{aligned} Q \ &\vdash \ Prf_T(\underline{\ulcorner R \urcorner}, \underline{\ulcorner G_T \urcorner}) \\ &\vdash \ \exists x Prf_T(x, \underline{\ulcorner R \urcorner}) \\ &\vdash \ \neg G_T. \end{aligned}$$

Hence $T \vdash \neg G_T$ because $T \supseteq Q$. But this contradicts the assumption that T is consistent. □

Claim: G_T is true (i.e., $N \models G_T$).

Proof: Since every logical consequence of Q is true, G_T says that there is no T-proof of G_T. The previous claim just showed this. □

So we have found G_T which is true but not a logical consequence of T. But we want G_T *and* $\neg G_T$ to be not logical consequences of T.

Claim: If T is a true theory (i.e., $T \subseteq \mathrm{Th}(N)$), then $T \nvdash \neg G_T$.

Proof: Obvious by the previous claim. □

Definition: T is *ω-consistent* if for any formula $A(x)$, if $T \vdash A(\underline{n})$ for all $n \in N$, then $T \nvdash \exists x \neg A(x)$ (i.e., $\forall x A(x)$ is consistent with T).

Claim: If T is ω-consistent, then $T \nvdash \neg G_T$.

Proof: Since $T \nvdash G_T$, $Q \vdash \neg Prf_T(\underline{n}, \underline{\ulcorner G_T \urcorner})$ for all $n \in N$. So $T \vdash \neg Prf_T(\underline{n}, \underline{\ulcorner G_T \urcorner})$, and hence

$$\begin{aligned} T \ &\nvdash \ \exists x \neg\neg Prf_T(x, \underline{\ulcorner G_T \urcorner}) \quad \text{by } \omega\text{-consistency} \\ &\nvdash \ \exists x Prf_T(x, \underline{\ulcorner G_T \urcorner}) \\ &\nvdash \ \neg G_T \end{aligned}$$

since $T \vdash \neg G_T \leftrightarrow \exists x Prf_T(x, \underline{\ulcorner G_T \urcorner})$. □

Theorem: If T is an ω-consistent axiomatizable extension of Q, then G_T is independent of T.[7]

Proof: By the above claims. □

Since G_T is independent of T, we might ask "what about the theory $T \cup \{G_T\}$?" This new theory will turn out to have another independent formula $G_{T \cup G_T}$. We can keep asking the question again and again and extend the theory:

$$T \cup \{G_T\} \cup \{G_{T \cup G_T}\} \cup \ldots \cup \{G_{T_\omega}\} \cup \{G_{T_\omega \cup G_{T_\omega}}\} \cup \ldots$$

Eventually, we get to a point where we can't recursively describe the ordinals.[8]

[7] And G_T is true. Even if it weren't true, $\neg G_T$ would be and would also be independent of T.

[8] $\omega + 1$ is recursively described by $a <_{\omega+1} b$ iff $b = 0 \wedge a \geq 1$ or $0 < a < b$; i.e. $<_{\omega+1}$ orders the integers as

$$1 <_{\omega+1} 2 <_{\omega+1} <_{\omega+1} \ldots <_{\omega+1} 0$$

Peano Arithmetic

Math 260C - Mathematical Logic

May 8, 1989

The crucial axiom of Peano arithmetic is the following induction axiom (paraphrased):

if $X \subseteq N$, $0 \in X$, and for all $n \in X$, $n+1 \in X$, then $X = N$.

We will try to give a first order axiomatizable theory that captures the induction axiom of Peano arithmetic. We won't be able to fully capture this notion because of the incompleteness theorem.

Definition: The language of PA, our theory for Peano arithmetic, is $0, S, +$, and $\cdot$. The axioms of PA are the seven axioms of Q plus all axioms of the form

$$\forall \vec{y}[(A(0,\vec{y}) \wedge \forall x(A(x,\vec{y}) \rightarrow A(Sx,\vec{y}))) \rightarrow \forall x A(x,\vec{y})]$$

for all formulas $A(x,\vec{y})$ with one or more free variables.

The difference between our set of axioms and Peano's induction axiom is that we can only state axioms for things that we can define while Peano's induction axiom doesn't say anything about the definability of the set X.

PA can prove things that Q can't.

Theorem: PA $\vdash \forall x \forall y(x+y = y+x)$.

Proof:

Claim 1: PA $\vdash \forall x \forall y(x + Sy = Sx + y)$.

Proof: Let $A(x,y)$ be $x + Sy = Sx + y$, and use induction on y.

a) PA $\vdash A(x,0)$ since

$$\begin{aligned} \text{PA} \vdash x + S0 &= S(x+0) && \text{by addition axiom} \\ &= Sx && \text{by addition axiom} \\ &= Sx + 0 && \text{by addition axiom.} \end{aligned}$$

b) PA $\vdash A(x,y) \to A(x,Sy)$ since[1]

$$\begin{aligned} \text{PA} \vdash x + S(Sy) &= S(x+Sy) && \text{by addition axiom} \\ &= S(Sx+y) && \text{by } A(x,y) \\ &= Sx + Sy && \text{by addition axiom.} \end{aligned}$$

So, by the induction axiom for $A(x,y)$ with respect to y,

$$\text{PA} \vdash \forall x \forall y (x + Sy = Sx + y). \ \Box$$

Claim 2: PA $\vdash \forall x(0 + x = x)$.

Proof: Let $B(x)$ be $0 + x = x$, and use induction on x.

a) PA $\vdash B(0)$ by an addition axiom.

b) PA $\vdash B(x) \to B(Sx)$ since

$$\begin{aligned} \text{PA} \vdash 0 + Sx &= S(0+x) && \text{by addition axiom} \\ &= Sx && \text{by } B(x). \end{aligned}$$

So by the induction axiom for $B(x)$ with respect to x,

$$\text{PA} \vdash \forall x(0 + x = x). \ \Box$$

Now let $C(x,y)$ be $x + y = y + x$, and use induction on y.

a) PA $\vdash C(x,0)$ by claim 2.

b) PA $\vdash C(x,y) \to C(x,Sy)$ since

$$\begin{aligned} \text{PA} \vdash x + Sy &= S(x+y) && \text{by addition axiom} \\ &= S(y+x) && \text{by } C(x,y) \\ &= y + Sx && \text{by addition axiom} \\ &= Sy + x && \text{by claim 1.} \end{aligned}$$

[1] Recall that since x and y are free, PA $\vdash A(x,y) \to A(x,Sy)$ means PA $\vdash \forall x \forall y (A(x,y) \to A(x,Sy))$.

So, by the induction axiom for $C(x, y)$ with respect to y,

$$\text{PA} \vdash \forall x \forall y(x + y = y + x). \ \square$$

Similarly, PA can prove the commutativity of multiplication, associativity laws, distributivity, etc.

Theorem: PA $\vdash \forall x \forall y(x < y \vee x = y \vee y < x)$.[2]
Proof:

Claim 1: PA $\vdash x < y \leftrightarrow Sx < Sy$.

Proof:

$$\begin{aligned} x < y &\Leftrightarrow \exists w(Sw + x = y) \\ &\Leftrightarrow \exists w(S(Sw + x) = Sy) \\ &\Leftrightarrow \exists w(Sw + Sx = Sy) \\ &\Leftrightarrow Sx < Sy. \ \square \end{aligned}$$

(Note that Q can prove this.)

Claim 2: PA $\vdash \forall x(x = 0 \vee 0 < x)$.

Proof: Q $\vdash x \neq 0 \rightarrow \exists y(Sy = x)$, so either $x = 0$, or $Sy + 0 = x$; i.e., $0 < x$. □

Now let $A(x, y)$ be $x < y \vee x = y \vee y < x$, and use induction on y.

a) PA $\vdash \forall x A(x, 0)$ by claim 2.

b) PA $\vdash \forall x A(x, y) \rightarrow \forall x A(x, Sy)$ since if we pick x arbitrarily, there are two cases to consider:

1. $x = 0$. Use claim 2 to show that $0 < Sy$.[3]
2. $x = Sw$ for some w. Then by $\forall x A(x, y)$, $w < y \vee w = y \vee y < w$. And by claim 1, $x < Sy \vee x = Sy \vee Sy < x$.

[2] Recall that $t < s$ means $\exists w(Sw + t = s)$. Also, remember that we proved this trichotomy relationship in Q provided that at least one of x and y was a standard number.

[3] This is all we need since $\forall x A(x, y) \rightarrow \forall x A(x, Sy)$ is equivalent to $\neg \forall x A(x, y) \vee \forall x A(x, Sy)$.

So, by the induction axiom for $\forall x(Ax, y)$ with respect to y,

$$\text{PA} \vdash \forall x \forall y A(x, y).\ \Box$$

PA can prove other similar things; e.g.,

$$\begin{array}{lcl}\text{PA} & \vdash & \forall x \forall y(\neg(x < y \wedge y < Sx)), \\ \text{PA} & \vdash & \forall x \forall y(x < y \rightarrow x \neq y \wedge y \not< x), \text{ and} \\ \text{PA} & \vdash & \forall x \forall y(x = y \rightarrow x \not< y).\end{array}$$

Peano Arithmetic (cont.)

Math 260C - Mathematical Logic

May 12, 1989

Least Number Principle: For any formula $A(x)$[1]

$$\text{PA} \vdash \exists x A(x) \rightarrow \exists x(A(x) \wedge (\forall z < x)\neg A(z)).$$

Proof: We will show that this is a consequence of the induction axioms. Let $B(x)$ be the formula $(\forall z < x)\neg A(z)$. Then PA $\vdash B(0)$ since nothing is less than 0. Now,[2]

$$\text{PA} \vdash [\neg\exists x(A(x) \wedge (\forall z < x)\neg A(z)) \wedge B(x)] \rightarrow B(x+1)$$

since

$$B(x) \wedge \neg B(x+1) \rightarrow A(x) \wedge (\forall z < x)\neg A(z)$$

and since

$$\text{PA} \vdash z < x+1 \rightarrow z < x \vee z = x.$$

[1] There may be other free variables $\vec{y}$ in A, but they are irrelevant to the theorem and its proof. Asserting things like PA $\vdash \phi$ where ϕ involves A and where A may have other free variables $\vec{y}$, is the same as saying that PA $\vdash \forall\vec{y}\phi$.

[2] Intuitively, what this formula says is that if there is no least element that makes A true ($\neg\exists x(A(x) \wedge (\forall z < x)\neg A(z))$) and no matter what x is all elements less than x make A false ($B(x)$), then all elements less than $x+1$ make A false ($B(x+1)$). It would seem that if there is no least element that makes A true, then no matter what x is all elements less than x would make A false. However we could have the following non-standard structure:

$$\underbrace{0,1,\ldots)}_{N}\underbrace{(\ldots-1^{'},0^{'},1^{'},\ldots)}_{Z}$$

with $n < m^{'}$ for $n \in N$ and $m^{'} \in Z$, $A(n)$ false for $n \in N$, and $A(m^{'})$ true for $m^{'} \in Z$. Then there is no least element that makes A true, but there are elements x such that some elements less than x make A true.

By the induction axiom for B with respect to x,

$$\mathrm{PA} \vdash \neg\exists x(A(x) \wedge (\forall z < x)\neg A(z)) \rightarrow \forall x B(x),$$

and

$$\mathrm{PA} \vdash \forall x B(x) \rightarrow \neg\exists x A(x).$$

This is the contrapositive of the theorem, so we're finished. □

Collection (Replacement) Axiom: Let $A(x, y, \vec{z})$ be a formula. (Ignoring $\vec{z}$, we'll write A as $A(x, y)$.) Then

$$\mathrm{PA} \vdash (\forall x < u)\exists y A(x, y) \rightarrow \exists t(\forall x < u)(\exists y < t)A(x, y).$$

The idea is that $t = \max\{y(x) : x < u\} + 1$.
Proof: Use induction on u. Let $B(w)$ be $\exists t(\forall x < w)(\exists y < t)A(x, y)$. Then $\mathrm{PA} \vdash B(0)$ since nothing is less than 0. Now,

$$\mathrm{PA} \vdash (\forall x < u)\exists y A(x, y) \wedge B(w) \wedge w < u \rightarrow B(Sw),$$

since

$$A(Sw, y^{'}) \wedge (\forall x < w)(\exists y < t)A(x, y) \rightarrow (\forall x < Sw)(\exists y \leq t + y^{'} + 1)A(x, y).$$

(Note: the last bound $t + y^{'} + 1$ could be improved to $\max(t, y^{'} + 1)$.) By induction on $w \leq u \rightarrow B(w)$ (or on $B(w) \vee u < w$),

$$(\forall x < u)\exists y A(x, y) \rightarrow B(u). \ \square$$

(We'll see later that the collection axiom allows us to interchange adjacent bounded quantifiers in formulas.)

With the help of the least number principle and the collection axiom, we'll begin to show that sequence coding, the β function, and ultimately the primitive recursive functions can be defined in PA. We already know that these functions, being recursive, are representable in Q. What we want to show is that they are strongly representable. We'll just sketch out the ideas until we get enough power to define primitive recursion; then we'll actually prove that primitive recursion is definable.

Recall from the fall quarter that a key idea in coding sequences was representing the sequence $\langle a_0, \ldots, a_n \rangle$ by the number

$$p_0^{a_0+1} \cdot p_1^{a_1+1} \cdot \ldots \cdot p_n^{a_n+1}$$

where p_i is the $(i+1)^{\mathrm{st}}$ prime. The problems with representing sequences in PA are the p_i's and exponentiation. PA can talk about the $(i+1)^{\mathrm{st}}$ prime, but in a round-about way. Essentially, the idea is to prove that there are infinitely many primes and then use the least number principle to successively pick out the primes $p_0, p_1, \ldots$.[3] To get $p_i^{a_i+1}$ without using exponentiation, recall that we were able to define p_ℓ for $\ell < p - 1$. The idea we used was that

$$\frac{p^{\ell-1}}{p-1} \equiv \ell \bmod p - 1$$

could be used for p^ℓ. This can be defined in PA[4] and can then be used to define greatest common divisors, least common multiples, etc. Eventually,

[3] Recall Euclid's proof that there are infinitely many primes: assume that there are a finite number of primes; take the product of these and add 1; the result is either a prime greater than any in the original set or factorizable by a prime greater than any in the original set. The problem with proving this in PA is that in order to define the product of an arbitrary finite number of numbers we would need primitive recursion.

In PA, that there are infinitely many primes is stated as $\forall x \exists y (y > x \wedge \text{“}y \text{ is prime”})$. (The relation “y is prime” is definable in PA similarly to the way we defined it in the fall quarter.) An approach to proving this in PA is to first find a z such that for all i between 1 and x, $i | z$. Then take the least divisor greater than 1 of $z+1$, and continue as in Euclid's proof. The problem now is how to find z. (In Euclid's proof we took the product.) The solution is to let $C(x, z)$ be

$$(\forall y < x)(\exists a(y \cdot a = z \vee y = 0) \wedge z \neq 0).$$

Then use induction on $\exists z C(x, z)$ with respect to z,

$$\mathrm{PA} \vdash \exists C(0, x) \wedge \forall x(\exists z C(x, z) \rightarrow \exists z C(x+1, z)),$$

to get the desired z.

[4] Division, remainders, and mod can be handled in PA, but we can't talk about p^ℓ directly since we haven't defined exponentiation. However PA can define “x is a power of p”, and then $\frac{p^{\ell-1}}{p-1}$ can be expressed indirectly in terms of “x is a power of p” and $x - 1 = \alpha(p-1)$ for $\alpha \equiv \ell \bmod p - 1$. Then,

$$\mathrm{PA} \vdash (x - 1 = \alpha(p-1) \rightarrow (xp - 1 = \beta(p-1)$$

for $\beta \equiv \ell + 1 \bmod p - 1$. I.e., $xp - 1 = (\alpha p + 1)(p - 1)$ since $x - 1 = \alpha(p-1)$. So $\beta = \alpha p + 1 \equiv \ell + 1 \bmod p - 1$.

we can get a definition for the β function which is the same definition that we derived in the fall quarter.
PA can prove simple properties about Seq, β, $*$, Len, etc. E.g.,

$$\mathrm{PA} \vdash \beta(i, w * \alpha) = \begin{cases} \beta(i,w) & i \leq \mathrm{Len}(w) \\ a & i = \mathrm{Len}(w) \\ 0 & \text{otherwise.} \end{cases}$$

With sequence coding, PA can define factorial, exponentiation, and the *primitive recursive functions*.

Definition: A set of formulas is *closed under bounded quantification* iff whenever ϕ is in the set, then so are the formulas $\forall x(x < t \rightarrow \phi)$ and $\exists x(x < t \wedge \phi)$ where t is any term not involving x.

Definition: $\Sigma^0_0 = \Pi^0_0 = \Delta^0_0$ is the smallest set of formulas containing all atomic formulas and closed under $\wedge, \vee, \neg, \rightarrow$, and bounded quantification.[5] For all i,

$$\begin{aligned} \Pi^0_i &= \{\forall x\phi : \phi \in \Sigma^0_{i-1}\}, \text{ and} \\ \Sigma^0_i &= \{\exists x\phi : \phi \in \Pi^0_{i-1}\}. \end{aligned}$$

In particular, Σ^0_1 is the set of formulas of the form $\exists x_1 \ldots \exists x_k \phi$ for some $\phi \in \Delta^0_0$.

Theorem: The set of formulas which are equivalent in PA to Σ^0_i (respectively Π^0_i) is closed under $\wedge, \vee$, and bounded quantification.

Proof: Use induction on i. $\wedge$ and $\vee$ are obvious by prenex operations. Let $A \in \Sigma^0_i$. Then A is of the form $\exists y_1 \ldots y_k \phi$ where $\phi \in \Pi^0_{i-1}$. Existentially bounding A produces

$$(\exists x < t)\exists y_1 \ldots \exists y_k \phi$$

which, by prenex operations, is equivalent to the Σ^0_i-formula

$$\exists x \exists y_1 \ldots \exists y_k (x < t \wedge \phi).$$

[5] We could also define this set as the set of formulas *logically equivalent* to the smallest set of formulas containing ....

If A is universally bound then

$$\text{PA} \vdash (\forall x \leq t)\exists y_1 \ldots \exists y_k \phi \leftrightarrow \exists u_1 (\forall x \leq t)(\exists y_1 \leq u_1)\exists y_2 \ldots \exists y_k \phi$$

by the collection axiom. By exchanging existential quantifiers,

$$\text{PA} \vdash (\forall x \leq t)\exists y_1 \ldots \exists y_k \phi \leftrightarrow \exists u_1 (\forall x \leq t)\exists y_2 \ldots \exists y_k (\exists y_1 \leq u_1)\phi.$$

Using induction on k, we get

$$\text{PA} \vdash (\forall x \leq t)\exists y_1 \ldots \exists y_k \phi \leftrightarrow \exists u_1 \ldots \exists u_k (\forall x \leq t)(\exists y_1 \leq u_1)\ldots(\exists y_k \leq u_k)\phi.$$

Then,

$$\text{PA} \vdash (\forall x \leq t)\exists y_1 \ldots \exists y_k \phi \leftrightarrow \exists u (\forall x \leq t)(\exists y_1 \leq u)\ldots(\exists y_k \leq u)\phi.$$

since such a u would be the maximum of the u_i's. By the induction hypothesis,

$$(\forall x \leq t)(\exists y_1 \leq u_1)\ldots(\exists y_k \leq u_k)\phi$$

is equivalent to a Π^0_{i-1}-formula. So we have reduced a Σ^0_i-formula to a Π^0_{i-1}-formula preceded by an existential quantifier. This is in Σ^0_i by definition.[6] (The proof of the closure of Π^0_i-formulas is dual.) □

Theorem: Let $f(\vec{x}) = y$ be a primitive recursive function. Then there is a formula $A(\vec{x}, y)$ such that

1. $A(\vec{x}, y)$ defines the graph of f,
2. $\text{PA} \vdash \forall \vec{x} \exists! y A(\vec{x}, y)$, and
3. A is a Σ^0_1-formula.

(Although this is not part of the theorem, we'll see later that A also represents f. Also, we will assume without proof that the theorem holds for functions like β, Seq, Len, etc. To see that this is true, recall the representations of these functions from the fall quarter.)

Proof: This theorem is obvious for 0, S, and Id. For composition, let

[6] The crucial idea in this proof is that the collection axiom allows us to interchange bounded quantifiers.

$$f(\vec{z}) = g(h_1(\vec{z}, \ldots, h_k(\vec{z})).$$

By induction, assume that A_g, A_{h_1}, ..., A_{h_k} are as in the theorem for g, h_1, ..., h_k. Then $A_f(\vec{z}, x)$ is

$$\exists y_1 \ldots \exists y_k (\bigwedge_{i=1}^{k} A_{h_i}(\vec{z}, y_i) \wedge A_g(\vec{y}, x)).$$

$A_g, A_{h_1}, \ldots, A_{h_k} \in \Sigma_1^0$, and so is A_f since Σ_1^0-formulas are closed under conjunction. That

$$\mathrm{PA} \vdash \forall \vec{z} \exists ! x A(\vec{z}, x)$$

follows easily from the induction hypotheses.

If f is defined by primitive recursion, then

$$\begin{aligned} f(\vec{x}, 0) &= g(\vec{x}), \text{ and} \\ f(\vec{x}, m+1) &= h(\vec{x}, m, f(\vec{x}, m)). \end{aligned}$$

By induction, assume that A_g and A_h are as in the theorem for g and h. Then $A_f(\vec{x}, m, y)$ is

$$\begin{aligned} &\exists w[\mathrm{Len}(w) = m+1 \wedge \mathrm{Seq}(w) && (1) \\ &\wedge \ A_g(\vec{x}, \beta(1, w)) && (2) \\ &\wedge \ (\forall i < m) A_h(\vec{x}, i, \beta(i+1, w), \beta(i+2, w)) && (3) \\ &\wedge \ y = \beta(m+1, w)]. && (4) \end{aligned}$$

w is a sequence representing the set of values computed in each "step" of the primitive recursion. (1) is not really needed because the remaining conjuncts ensure that w is a sequence with the appropriate number of elements, but it is included for reasons of clarity. (2) represents the base case where $i = 0$; the value for this case is the first element of the sequence w. (3) represents all the other cases for $0 < i \leq m$; these values are stored in successive positions in the sequence w. This line of the definition of A_f represents the fact that each element in the sequence except the first is obtained from the preceding element; i.e., $h(\vec{x}, i, \beta(i+1, w)) = \beta(i+2, w)$. (4) represents the result of the primitive recursion; i.e., the final value in the sequence.

$A_f \in \Sigma^0_1$ because A_h is boundedly quantified, β is PA-equivalent to a Σ^0_1-formula, and Len and Seq are assumed to be Σ^0_1-formulas. So the whole formula is a conjunction of Σ^0_1 formulas preceded by an existential quantifier.

In order to show uniqueness; i.e., that

$$\mathrm{PA} \vdash \forall \vec{x} \exists ! y (A_f(\vec{x}, m, y)),$$

we would first prove by induction up to m that the w of the definition of A_f exists. Then we would prove by induction that each element of w is unique. I.e., for two sequences, we would use existence to show that the j^{th} elements exist, and then use uniqueness of the j^{th} elements to show the uniqueness of the $(j+1)^{\text{st}}$ elements. □

Proof Predicates in PA

Math 260C - Mathematical Logic

May 15, 1989

We're headed towards Gödel's second incompleteness theorem which says that for any "sufficiently strong" axiomatizable theory T, $T \nvdash \mathrm{Con}(T)$; i.e., T can not prove its own consistency. We'll do this in PA by developing the relation 'Con' and then proving that PA $\nvdash$ Con(PA).[1] In PA, we'll develop formulas which say things about the Gödel numbers of proofs, and we'll show that PA can not prove its own consistency; i.e., PA $\nvdash$ "$0 = 1$ is not a theorem of PA". Our first goal will be "the arithmetization of meta-mathematics" (i.e., the Gödel numbering of the syntax of first order logic).

Example: Recall the Num function, Num : $n \mapsto \ulcorner \underline{n} \urcorner$ $(= \ulcorner S^n 0 \urcorner)$. This is primitive recursive and can be defined in PA with a Σ^0_1-definition; i.e.,

$$\begin{aligned} \ulcorner \underline{n} \urcorner &= \ulcorner S(S(\ldots(S(0))\ldots))\urcorner \\ &= 6161\ldots61522\ldots22. \end{aligned}$$

With this definition, PA can prove that

$$\mathrm{Num}(x+1) = 61 \cdot 10^{\lceil \log_{10} \mathrm{Num}(x) \rceil} + 10 \cdot \mathrm{Num}(x) + 2.$$

PA can also prove that $\forall x \exists y(\mathrm{Num}(x) = y)$.[2]

We won't go into detail how the following functions and predicates are defined in PA; rather, we'll just note that each of them can be viewed as

[1] We could also do this in Q; i.e., show that Q $\nvdash$ Con(Q). The reason for not doing so is that it is more difficult. For Q, it can be done by constructing a non-standard model $\mathcal{Q}$ of Q such that $\mathcal{Q} \models \neg\mathrm{Con}(\mathrm{Q})$. Similarly, we could show that ZF $\nvdash$ Con(ZF), although again this is harder than in PA. Note that for Gödel's first incompleteness theorem, we showed, by virtue of Q's ω-inconsistency, that Q $\nvdash$ "I am not provable".

[2] Note that since PA can prove this, it is true even for non-standard models of PA. The y that equals Num(x) for non-standard x will itself be non-standard; i.e., it will consist of a non-standard number of 61's followed by a 5 followed by a non-standard number of 2's.

syntactic functions or predicates which could be defined if we felt like doing the work.

Term(x) $\Leftrightarrow$ "x is the Gödel number of a term".

Digit(i, x) $=$ the i^{th} digit in base 10 of x.

Symbol(i, x) $=$ the i^{th} logical symbol in x viewed as a Gödel number.

Concat(x, y) $=$ the base 10 concatenation of x and y.

A-formula(x) $\Leftrightarrow$ "x is the Gödel number of an atomic formula".

Formula(x) $\Leftrightarrow$ "x is the Gödel number of a formula".

Free(x, y) $\Leftrightarrow$ Formula(x) $\wedge$ "y is the Gödel number of a variable occurring free in x".

PA-axiom(x) $\Leftrightarrow$ "x is the Gödel number of an axiom of PA".

Sub(x, y, z) $= \ulcorner A(\underline{z}/y) \urcorner$ if $x = \ulcorner A \urcorner$, and y is the Gödel number of a variable.

$\mathbf{Proof_{PA}}(x)$ $\Leftrightarrow$ "x is the Gödel number of a valid PA proof".[3]

$\mathbf{Prf_{PA}}(x, y)$ $\Leftrightarrow$ "x is the Gödel number of a PA proof of the formula whose Gödel number is y".

The above functions and predicates are all primitive recursive (and also representable in Q). This is what is meant by the arithmetization of metamathematics. The following predicate is not primitive recursive (and in fact not even decidable):

$\mathbf{Thm_{PA}}(y)$ $\Leftrightarrow \exists x \mathrm{Prf}_{\mathrm{PA}}(x, y)$.

With the arithmetization functions and predicates defined above, our next goal will be to show that if PA can prove the sentence A, then PA can prove that A is a theorem; i.e.,

[3] More generally,

$\mathbf{Proof_{T}}(x)$ $\Leftrightarrow$ "x is the Gödel number of a valid proof in the theory T which has a definable set of axioms".

$$\text{if PA} \vdash A, \text{ then PA} \vdash \mathrm{Thm}_{\mathrm{PA}}(\underline{\ulcorner A \urcorner}).$$

Also, we'll show that PA can prove that if A is a theorem, then the fact that it is a theorem is also a theorem; i.e.,

$$\mathrm{PA} \vdash \mathrm{Thm}_{\mathrm{PA}}(\underline{\ulcorner A \urcorner}) \to \mathrm{Thm}_{\mathrm{PA}}(\underline{\ulcorner \mathrm{Thm}_{\mathrm{PA}}(\underline{\ulcorner A \urcorner}) \urcorner}).$$

Theorem 1: If A is a Σ^0_1-sentence and $\mathrm{N} \models A$, then $\mathrm{PA} \vdash A$.[4]

Theorem 2: If $A(\vec{x})$ is a Σ^0_1-formula, then[5]

$$\mathrm{PA} \vdash A(\vec{x}) \to \exists w \mathrm{Prf}_{\mathrm{PA}}(w, \underline{\ulcorner A(\underline{x_1}, \ldots, \underline{x_k}) \urcorner}).$$

Proof: of theorems 1 and 2.[6] Use induction on the complexity of A.

> *Lemma:* If $t(\vec{x})$ is a term, then
>
> $$\mathrm{PA} \vdash \underline{t(\vec{x})} = t(\underline{n_1}, \ldots, \underline{n_k}) \qquad (1)$$
>
> for all $\vec{n} \in N^k$, and
>
> $$\mathrm{PA} \vdash \forall x \exists w \mathrm{Prf}_{\mathrm{PA}}(w, \ulcorner \underline{t(\vec{x})} = t(\underline{x_1}, \ldots, \underline{x_k}) \urcorner). \qquad (2)$$
>
> *Proof*: (Sketch.) We have already proved (1) for Q since $S, +$, and $\cdot$ were representable. In that proof, we used induction outside of Q to reason about Q. For the second half of the lemma, we would formalize in PA the argument we used to prove the first half. We would use induction inside PA to talk about Q.

(Proof of theorems 1 and 2 continued.) Without loss of generality, assume that all negation signs in A are in front of atomic formulas.

Basis: A is atomic or negated atomic. Then $A(\vec{x})$ is either $s(\vec{x}) = t(\vec{x})$ or $s(\vec{x}) < t(\vec{x})$.

[4] The theorem is also true if we replace $\mathrm{PA} \vdash A$ by $\mathrm{Q} \vdash A$.

[5] The theorem is also true if we replace $\mathrm{Prf}_{\mathrm{PA}}$ by $\mathrm{Prf}_{\mathrm{Q}}$. But *not* if we replace $\mathrm{PA} \vdash \ldots$ by $\mathrm{Q} \vdash \ldots$.

[6] Intuitively, for the proof of theorem 1, since A is a Σ^0_1-sentence, A asserts the existence of something. Any non-standard model of PA has N inside it. So if $\mathrm{N} \models A$, then $\mathrm{PA} \vdash A$.

For theorem 1, we have already proved that if $s(\vec{n}) = t(\vec{n})$, then $\mathrm{Q} \vdash \underline{s(\vec{n})} = \underline{t(\vec{n})}$ since $=$ is representable and since $\mathrm{Q} \vdash s(\underline{n_1}, \ldots, \underline{n_k}) = \underline{s(\vec{n})}$. Similarly for $<$.

For theorem 2, we formalize this reasoning to show that

$$\mathrm{PA} \vdash s(\vec{x}) < t(\vec{x}) \to \mathrm{Thm}_{\mathrm{PA}}(\ulcorner S(\underline{x_1}, \ldots, \underline{s_k}) < t(\underline{x_1}, \ldots, \underline{x_k})\urcorner).$$

(Proof continued next time.)

Proof Predicates, Gödel's Second Incompleteness Theorem

Math 260C - Mathematical Logic

May 19, 1989

Last time, we started proving the following two theorems.

Theorem 1: If A is a Σ_1^0-sentence and $\mathrm{N} \models A$, then $\mathrm{PA} \vdash A$.

Theorem 2: If $A(\vec{x})$ is a Σ_1^0-formula, then

$$\mathrm{PA} \vdash A(\vec{x}) \rightarrow \mathrm{Thm}_{\mathrm{PA}}(\underline{\ulcorner A(\underline{x_1}, \ldots, \underline{x_k})\urcorner}).$$

The proofs are by induction on the logical complexity of A. They depend on induction up to n and on the formalizability of proofs.

Proof of 1: Without loss of generality, assume that A contains no $\rightarrow$ symbols and that negation symbols apply only to atomic formulas. The proof will actually show that if A is true, then $\mathrm{Q} \vdash A$.
Basis: A is atomic or negated atomic. Then if A is true, $\mathrm{Q} \vdash A$ since $<, =, S, 0, *$, and $\cdot$ are all representable in Q.
Induction: There are three cases to consider.

1. A is $B \wedge C$ or is $B \vee C$. If A is $B \wedge C$, then by induction, $\mathrm{Q} \vdash B$ and Q $\vdash C$. So we can concatenate the proofs of B and C to get $\mathrm{Q} \vdash B \wedge C$. If A is $B \vee C$, then by induction, $\mathrm{Q} \vdash B$ or $\mathrm{Q} \vdash C$. So we can pick one of the proofs of B and C to get $\mathrm{Q} \vdash B \vee C$.

2. A is $(\forall x < t)B(x)$. Then t has no free variables, and by a lemma from last lecture, there is an $n \in N$ such that $\mathrm{Q} \vdash \underline{n} = t$ since $0, S, +$, and $\cdot$ are representable in Q. Now

$$\mathrm{Q} \vdash x < \underline{n} \rightarrow x = \underline{0} \vee x = \underline{1} \vee \ldots \vee x = \underline{n-1}$$

by induction on n. So by the induction hypothesis and since A is true, $\mathrm{Q} \vdash B(\underline{m})$ for $m = 0, \ldots, n-1$. Combining these $m+1$ proofs, we get $\mathrm{Q} \vdash A$.[1]

3. A is $\exists x B(x)$. ($(\exists x < t)B(x)$ is handled similarly.) Since A is true, there is an $n \in N$ such that $B(\underline{n})$ is true. By induction, $\mathrm{Q} \vdash B(\underline{n})$, so $\mathrm{Q} \vdash \exists x B(x)$.[2] □

Proof of 2: Formalizing the proof of theorem 1 inside PA shows that

$$\mathrm{PA} \vdash A \to \mathrm{Thm}_{\mathrm{Q}}(\underline{\ulcorner A \urcorner}).$$

Hence

$$\mathrm{PA} \vdash A \to \mathrm{Thm}_{\mathrm{PA}}(\underline{\ulcorner A \urcorner}). \ \square$$

Corollary 1: Let A be any sentence. If $\mathrm{PA} \vdash A$, then $\mathrm{PA} \vdash \mathrm{Thm}_{\mathrm{PA}}(\underline{\ulcorner A \urcorner})$.

Proof: Take B to be the sentence $\mathrm{Thm}_{\mathrm{PA}}(\underline{\ulcorner A \urcorner})$ and apply theorem 1 to B. □

Corollary 2: Let A be any sentence. Then

$$\mathrm{PA} \vdash \mathrm{Thm}_{\mathrm{PA}}(\underline{\ulcorner A \urcorner}) \to \mathrm{Thm}_{\mathrm{PA}}(\underline{\ulcorner \mathrm{Thm}_{\mathrm{PA}}(\underline{\ulcorner A \urcorner}) \urcorner}).$$

Proof: Take B to be the sentence $\mathrm{Thm}_{\mathrm{PA}}(\underline{\ulcorner A \urcorner})$ and apply theorem 2 to B. □

In order to apply theorems 1 and 2 to B in the above two corollaries, we need $B \in \Sigma_1^0$. So,

[1] An implicit assumption that we make is that we have a nice way of combining proofs to get a single proof. We don't need to be able to do this here, but we will in the proof of theorem 2. In fact, since the only class of functions that we can represent in PA are the primitive recursive functions, we need to be able to formalize the fact that there is a primitive recursive way of combining proofs in PA.

[2] More explicitly, using the Kleene proof system, the Q-proof of $\exists x B(x)$ is constructed as follows:

$\vdots$	
$B(\underline{n})$	proof assumed by induction
$B(\underline{n} \to \exists x B(x)$	axiom
$\exists x B(x)$	by modus ponens

Theorem: [3] If R is a k-ary primitive recursive predicate, then there is a Σ^0_1-formula $A(\vec{x})$ and a Π^0_1-formula $B(\vec{x})$ such that

1. $\mathrm{PA} \vdash A(\vec{x}) \leftrightarrow B(\vec{x})$, and
2. for all $\vec{n} \in N^k$, $R(\vec{n})$ iff $\mathrm{N} \models A(\vec{n})$ (iff $\mathrm{N} \models B(\vec{n})$).

Proof: The characteristic function for R, $\chi_{\mathrm{R}}(\vec{x})$, is primitive recursive and so is Σ^0_1-definable in PA by a formula $C(\vec{x}, y)$ such that $\mathrm{PA} \vdash \forall\vec{x}\exists! y C(\vec{x}, y)$. Let $A(\vec{x})$ be $C(\vec{x}, 1)$, and let $B(\vec{x})$ be $\forall y(y \neq 1 \rightarrow \neg C(\vec{x}, y)$. (Equivalently, let $B(\vec{x})$ be $\forall y(y = 1 \vee \neg C(\vec{x}, y)$.) Then $\chi_{\mathrm{R}} = 1$ iff R is true, and $A \in \Sigma^0_1$, and B is equivalent to a Π^0_1-formula. □

In particular, $\mathrm{Proof}_{\mathrm{PA}}$ is equivalent to a Σ^0_1-sentence or a Π^0_1-sentence.

Corollary: $\mathrm{Thm}_{\mathrm{PA}}(x)$ is a Σ^0_1-formula.

Proof: $\mathrm{Thm}_{\mathrm{PA}}(x)$ is $\mathrm{Proof}_{\mathrm{PA}}(w, x)$ and $\mathrm{Proof}_{\mathrm{PA}}(-,-)$ is (equivalent to) a Σ^0_1-formula. □

Definition: Let T be a theory in the language of Q, and let $B(x)$ be a formula. B is a *provability predicate* for T iff for all sentences A and C,

PP-0) T is an extension of Q (so that T can represent Diag),

PP-1) if $\mathrm{T} \vdash A$, then $\mathrm{T} \vdash B(\underline{\ulcorner A \urcorner})$,

PP-2) $\mathrm{T} \vdash B(\underline{\ulcorner A \urcorner}) \wedge B(\underline{\ulcorner A \rightarrow C \urcorner}) \rightarrow B(\underline{\ulcorner C \urcorner})$, and

PP-3) $\mathrm{T} \vdash B(\underline{\ulcorner A \urcorner}) \rightarrow B(\underline{\ulcorner B(\underline{\ulcorner A \urcorner}) \urcorner})$.

Claim: $\mathrm{Thm}_{\mathrm{PA}}$ is a provability predicate for PA.

Proof: PP-0 is obviously satisfied. PP-1 and PP-3 are satisfied by corollaries 1 and 2 above. And PP-2 is satisfied since we can concatenate the proofs of A and $A \rightarrow C$ and get C by modus ponens.[4] □

[3] This is a restatement for predicates of an earlier theorem which showed that primitive recursive functions were Σ^0_1-definable.

[4] I.e.,

$$\begin{array}{ll}
\vdots & \\
A & \text{proof of } A \\
\vdots & \\
A \rightarrow C & \text{proof of } A \rightarrow C \\
C & \text{by modus ponens}
\end{array}$$

Gödel's Second Incompleteness Theorem: If B is a provability predicate for T,[5] then

$$\mathrm{T} \nvdash \neg B(\underline{\ulcorner 0 = 1 \urcorner}),$$

and

$$\mathrm{T} \nvdash \neg \exists x(\text{“}x \text{ is the Gödel number of a sentence”} \wedge \neg B(x)).$$

Proof: Later.

Definition: Let T be any theory. Con(T) is $\neg \mathrm{Thm}_{\mathrm{T}}(\underline{\ulcorner 0 = 1 \urcorner})$. I.e., Con(T) is the assertion that T is consistent. In particular, Con(PA) is $\neg \mathrm{Thm}_{\mathrm{PA}}(\underline{\ulcorner 0 = 1 \urcorner})$.

Corollary: PA $\nvdash$ Con(PA).

The second incompleteness theorem puts a limit on what can be proved; e.g., PA can not prove its own consistency. If we create a new theory, say PA^+, by adding Con(PA) to PA, then $\mathrm{PA}^+ \vdash \mathrm{Con(PA)}$, but $\mathrm{PA}^+ \nvdash \mathrm{Con}(\mathrm{PA}^+)$. So

Corollary: If T is any consistent axiomatizable extension of PA, then T $\nvdash$ Con(T).

T is inconsistent iff $\mathrm{T} \vdash 0 = 1$, since if T is consistent then $\mathrm{T} \vdash 0 \neq 1$. Now suppose that $\mathrm{T} \vdash \mathrm{Thm}_{\mathrm{T}}(\underline{\ulcorner A \urcorner})$. Does this imply that $\mathrm{T} \vdash A$? No. Take T to be PA $\cup\{\neg\mathrm{Con(PA)}\}$. T is consistent since T $\supset$PA, and PA $\nvdash$ Con(PA). Now $\mathrm{T} \vdash \mathrm{Thm}_{\mathrm{PA}}(\underline{\ulcorner 0 = 1 \urcorner})$, and so $\mathrm{T} \vdash \mathrm{Thm}_{\mathrm{T}}(\underline{\ulcorner 0 = 1 \urcorner})$ because any PA-proof is a T-proof. But $\mathrm{T} \nvdash 0 = 1$ since T is consistent. In any model for T, the w that exists by virtue of $\exists w \mathrm{Proof}_{\mathrm{T}}(w, \underline{\ulcorner 0 = 1 \urcorner})$ will be non-standard.

Proof of Gödel's Second Incompleteness Theorem: (Think of T as PA and B as $\mathrm{Thm}_{\mathrm{PA}}$.) By the diagonalization lemma, there is a sentence G such that

$$\mathrm{T} \vdash G \leftrightarrow \neg B(\underline{\ulcorner G \urcorner}).$$

The following two claims suffice to prove the theorem.

Claim 1: $\mathrm{T} \vdash G \leftrightarrow \mathrm{Con(T)}$.

Proof:

[5] This will work for any axiomatizable theory extending Q.

1. $\mathrm{T} \vdash \neg G \to B(\underline{\ulcorner G \urcorner})$ by definition of G.
2. $\mathrm{T} \vdash \neg G \to B(\underline{\ulcorner B(\underline{\ulcorner G \urcorner}) \urcorner})$ since $\mathrm{T} \vdash B(\underline{\ulcorner G \urcorner}) \leftrightarrow B(\underline{\ulcorner B(\underline{\ulcorner G \urcorner}) \urcorner})$ by PP-3 and by (1).
3. $\mathrm{T} \vdash B(\underline{\ulcorner B(\underline{\ulcorner G \urcorner}) \to \neg G \urcorner})$ by PP-1 and since $\mathrm{T} \vdash B(\underline{\ulcorner G \urcorner}) \to \neg G$.
4. $\mathrm{T} \vdash \neg G \to B(\underline{\ulcorner \neg G \urcorner})$ by (2), (3), and PP-2.
5. $\mathrm{T} \vdash \neg G \to B(\underline{\ulcorner 0 = 1 \urcorner})$ by using PP-1 to get $\mathrm{T} \vdash B(\underline{\ulcorner G \to (\neg G \to 0 = 1) \urcorner})$ since $G \to (\neg G \to 0 = 1)$ is valid, and by (1), (4), and two uses of PP-2.
6. $\mathrm{T} \vdash \neg G \to \neg \mathrm{Con}(\mathrm{T})$ by (1), (4), and PP-2.
7. $\mathrm{T} \vdash B(\underline{\ulcorner 0 = 1 \urcorner}) \to B(\underline{\ulcorner G \urcorner})$ by PP-2 and since $\mathrm{T} \vdash B(\underline{\ulcorner 0 = 1 \to G \urcorner})$ by PP-1.
8. $\mathrm{T} \vdash G \to \mathrm{Con}(\mathrm{T})$ by using G, the definition of G, and the definition of $\mathrm{Con}(\mathrm{T})$.[6] □

Claim 2: $\mathrm{T} \nvdash G$.

Proof: Suppose that $\mathrm{T} \vdash G$. Then $\mathrm{T} \vdash \neg B(\underline{\ulcorner G \urcorner})$ by the definition of G. By PP-1, $\mathrm{T} \vdash B(\underline{\ulcorner G \urcorner})$ since G is a theorem. But then T is inconsistent, a contradiction. □

This theorem killed Hilbert's program by saying that in order to use a theory, the consistency of the theory must be established *outside* of the theory. Even if we could prove that the theory was consistent inside the theory, we would still be implicitly depending on the consistency of the theory. (I.e., if the theory was inconsistent, it could prove anything.) Hilbert wanted to prove the consistency of set theory inside a weaker system (PA).

Gentzen proved that if ϵ_0[7] is well founded, then PA is consistent. Typically, by well founded, we mean well ordered. So what Gentzen showed is

[6] So G, which says "I'm not provable", is equivalent to saying that T is consistent. In PA, we would have

$$\mathrm{PA} \vdash \neg G \to \mathrm{Thm}_{\mathrm{PA}}(\underline{\ulcorner \neg G \urcorner})$$

since $\neg G$ is equivalent to a Σ^0_1-formula.

[7] ϵ_0 is an ordinal. It is built up from ω as follows:

$$\begin{aligned} &\omega \\ \omega \cdot 2 &= \omega + \omega \end{aligned}$$

that if any primitive recursive subset of ϵ_0 has a least element, then PA is consistent. Armed with our knowledge of incompleteness, what this tells us is that although PA can formalize the notion that any primitive recursive subset of ϵ_0 has a least element, PA can't prove that this notion is true. In order to believe that PA is consistent, all we must believe is that ϵ_0 is well-founded. ϵ_0 characterizes exactly the consistency of PA.

More precisely, ϵ_0 characterizes exactly the 1-consistency of PA where the 1-consistency of PA is defined as the following set of formulas:[8]

$$\{\mathrm{Thm}_{\mathrm{PA}}(\ulcorner \underline{A} \urcorner) \to A : A \in \Sigma^0_1\}.$$

The 1-consistency of PA asserts that every existential statement that is provable in PA is true. What Gentzen showed was that the 1-consistency of PA is equivalent to the well-foundedness of ϵ_0 in PA.

$$\begin{array}{rcl}
\omega \cdot 3 & = & \omega \cdot 2 + \omega \\
 & \vdots & \\
\omega^2 & = & \lim\limits_{n\to\infty} \omega \cdot n \\
\omega^3 & = & \lim\limits_{n\to\infty} \omega^2 \cdot n \\
 & \vdots & \\
\omega^\omega & = & \lim\limits_{n\to\infty} \omega^n \quad (= \bigcup\limits_n \omega^n) \\
\omega^{\omega^\omega} & = & \lim\limits_{n\to\infty} \omega^{\omega^n} \\
 & \vdots & \\
\omega \Uparrow \omega & = & \lim\limits_{n\to\infty} \omega \Uparrow n \;=\; \epsilon_0.
\end{array}$$

[8] The '1' in '1-consistency' corresponds to the '1' in 'Σ^0_1'. Similarly, we could define the 2-consistency of PA, etc.

Löb's Theorem, Arithmetic Hierarchy

Math 260C - Mathematical Logic

May 22, 1989

Löb's theorem relates to the distinction we encountered last time between a formula being provable and being true. Before we get to Löb's theorem, we need the formalized version of the deduction theorem which formalizes the statement "if PA $\cup\{A\} \vdash B$, then PA $\vdash (A \to B)$".

Formalized Version of the Deduction Theorem: Let A and B be sentences. Then

$$\mathrm{PA} \vdash \mathrm{Thm}_{\mathrm{PA}\cup\{\mathrm{A}\}}(\ulcorner \underline{B} \urcorner) \to \mathrm{Thm}_{\mathrm{PA}}(\ulcorner \underline{A \to B} \urcorner).$$

Proof: (Note that the converse is obvious.) Using Kleene style proofs, we have a sequence of formulas $C_1, C_2, \ldots, C_t = B$ that prove B where the axioms are PA $\cup\{A\}$. To get a proof of $A \to B$ using only the axioms of PA, we build the sequence of formulas $\ldots, A \to C_1, \ldots, A \to C_2, \ldots, A \to C_t$.[1] □

Löb's Theorem: Let A be any sentence. If PA $\vdash \mathrm{Thm}_{\mathrm{PA}}(\ulcorner \underline{A} \urcorner) \to A$, then PA $\vdash A$.[2]

[1] The "..." before each $A \to C_j$ represent some sub-formulas leading up to $A \to C_j$. For example, to get $A \to C_j$ in the new sequence from C_j in the old sequence where C_j is an axiom, we really have

$$\frac{C_j \quad C_j \to (A \to C_j)}{A \to C_j}$$

and if C_j is derived by modus ponens from C_i and $C_i \to C_j$, then we reall have

$$\frac{A \to C_i \quad A \to (C_i \to C_j)}{A \to C_j}$$

It is also possible to prove the formalized version of the deduction theorem using refutation style proofs by proving that a refutation of B in PA $\cup \{A\}$ can be transformed into a refutation of $A \to B$ in PA.

[2] Löb's theorem actually holds for any theory T extending Q and any provability predicate for T.

Proof: Suppose that PA $\not\vdash A$. Then let S be PA $\cup\{\neg A\}$. S is a consistent theory since PA $\not\vdash A$. Now

$$\mathrm{S} \vdash \neg\mathrm{Thm}_{\mathrm{PA}}(\underline{\ulcorner A \urcorner}),$$

since PA $\vdash \mathrm{Thm}_{\mathrm{PA}}(\underline{\ulcorner A \urcorner}) \rightarrow A$. So

$$\mathrm{S} \vdash \neg\mathrm{Thm}_{\mathrm{PA}}(\underline{\ulcorner \neg A \rightarrow 0 = 1 \urcorner})$$

since PA $\vdash 0 \neq 1$. Thus, by the formalized version of the deduction theorem, (with A as in that theorem)

$$\mathrm{S} \vdash \neg\mathrm{Thm}_{\mathrm{S}}(\underline{\ulcorner 0 = 1 \urcorner}).$$

I.e., S $\vdash$ Con(S). But this contradicts the second incompleteness theorem. So PA $\vdash A$. □

In terms of models, Löb's theorem says that if there is a (non-standard) proof of A which implies that A is true, then there is a standard proof of A.

Arithmetic Hierarchy

Theorem: Let R be a k-ary predicate. Then R is r.e. iff R is defined by some Σ^0_1-formula $A(x_1, \ldots, x_k)$; i.e., for all $\vec{n} \in N^k$, $\vec{n} \in R$ iff $\mathrm{N} \models A(\underline{\vec{n}})$.

Proof: $\Rightarrow$: By the Kleene normal form theorem (from fall quarter) there is a primitive recursive predicate $S(\vec{x}, y)$ such that for all $\vec{n} \in N^k$,

$$R(\vec{n} \Leftrightarrow \exists y S(\vec{n}, y).$$

And S is defined by a Σ^0_1-formula.
$\Leftarrow$: Suppose $A(x_1, \ldots, x_k)$ is $\exists y_1 \ldots \exists y_\ell B(\vec{x}, \vec{y})$ with $B \in \Delta^0_0$. Then

$$\mathrm{N} \models A(n_1, \ldots, n_k) \Leftrightarrow \exists y^* \mathrm{N} \models B(n_1, \ldots, n_k, \beta(1, y^*), \ldots, \beta(\ell, y^*)).$$

But $B(\vec{x}, \vec{y})$ and β are primitive recursive. So $\{\vec{n} : A(\vec{n})\}$ is r.e. □

Generalizing this result yields the arithmetic hierarchy:

$$\begin{tikzcd}
 & \vdots & \\
 & \Delta_3^{\mathrm{K}} & \\
\Pi_3^{\mathrm{K}} \arrow[ur, "\subseteq"] & & \Sigma_3^{\mathrm{K}} \arrow[ul, "\supseteq"'] \\
 & \Delta_2^{\mathrm{K}} \arrow[ul, "\supseteq"'] \arrow[ur, "\subseteq"] & \\
\Pi_2^{\mathrm{K}} \arrow[ur, "\subseteq"] & & \Sigma_2^{\mathrm{K}} \arrow[ul, "\supseteq"'] \\
 & \Delta_1^{\mathrm{K}} \arrow[ul, "\supseteq"'] \arrow[ur, "\subseteq"] & \\
\Pi_1^{\mathrm{K}} \arrow[ur, "\subseteq"] & & \Sigma_1^{\mathrm{K}} \arrow[ul, "\supseteq"'] \\
 & \Delta_0^{\mathrm{K}} \arrow[ul, "\supseteq"'] \arrow[ur, "\subseteq"] &
\end{tikzcd}$$

This is a hierarchy of predicates defined by formulas.[3] In general,

$$\begin{aligned}
\Delta_i^{\mathrm{K}} &= \Sigma_{i+1}^{\mathrm{K}} \cap \Pi_{i+1}^{\mathrm{K}}, \\
\Sigma_i^{\mathrm{K}} &= \{R \subseteq N^\ell : R \text{ is definable by a } \Sigma_i^0\text{-formula}\}, \text{ and} \\
\Pi_i^{\mathrm{K}} &= \{R \subseteq N^\ell : R \text{ is definable by a } \Pi_i^0\text{-formula}\}.
\end{aligned}$$

In particular, Δ_0^{K} is the set of recursive predicates, Σ_1^{K} is the set of r.e. predicates, and Π_1^{K} is the set of co-r.e. predicates. The union of all of the sets in the arithmetic hierarchy is the set of all predicates definable in $(N, 0, S, +, \cdot)$.

[3] The superscript K's in the above symbols represent the fact that that the predicates are defined by the Kleene normal form theorem. This notation is *not* standard.

Matiyasevich's Theorem: [4] Every r.e. set can be defined by a Σ_1-formula (purely existential, no bounded quantifiers).

This theorem showed that Hilbert's tenth problem, that of finding a method for deciding whether a given diophantine equation has a solution, is not possible.

Definition: A *diophantine equation* is an equality between two multi-variable polynomials over integers.

What Matiyasevich showed was that any r.e. predicate can be expressed as $\exists\vec{y}(p(\vec{x},\vec{y}) = 0)$ where p is a multi-variable polynomial over the integers.

[4] Matiyasevich's theorem is built upon earlier work by Davis, Putnam, and Robinson.

Introduction to Modal Logic

Math 260C - Mathematical Logic

May 22, 1989

Modal logic extends propositional and first order logic with a new symbol, $\Box$, the necessitation operator. $\Box A$ means that A is necessary or that A must be true; as opposed to contingent truths which just happen to be true. $\Box A$ can have many different interpretations:

- "A is logically true"; i.e., true based on pure logic.
- "A is a theorem of PA"; e.g., Löb's axiom:

$$\Box(\Box A \to A) \to \Box A.$$

- "A is a consequence of the laws of physics"; i.e., water freezes at 0° C at standard temperature and pressure, as opposed to water in the freezer is frozen.
- "A is known to be true."
- "A is believed to be true."
- "A ought to be true."
- "A will be true at all times in the future."
- "A will be true when the program halts."

Modal Logic - Syntax, Models, Proof Theory, Soundness and Completeness

Math 260C - Mathematical Logic

May 26, 1989

In propositional modal logic, we have the following symbols:

Propositional Variables: $p_1, p_2, \ldots$.

Propositional Connectives: $\wedge, \vee, \neg, \rightarrow$.

Modal Connective: $\Box$.

Punctuation: ().

The well-formed formulas, wff's, are inductively defined by

1. p_i is a wff.
2. If A and B are wff's then $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$, $\neg A$, and $\Box A$ are wff's.

$\Diamond$ is an abbreviation for $\neg\Box\neg$. $\Diamond A$ means $\neg\Box\neg A$; i.e., "A is possible".

Definition: A *propositional truth valuation* is a mapping

$$\sigma : \{\text{wff's}\} \rightarrow \{T, F\}$$

which respects the propositional connectives, but treats formulas of the form $\Box A$ as atomic formulas.

In other words, a propositional truth valuation σ, is determined by its values on propositional variables and on formulas of the form $\Box A$ by the usual meanings of propositional connectives. (σ views $\Box A$ as a propositional variable.) So it is possible to have

$$\sigma(\Box A) \neq \sigma(\Box(A \vee A)).$$

But “good” propositional truth valuations (to be defined later) will not behave this way.

Kripke Models

A Kripke model is a triple $(\mathcal{K}, R, \phi)$ with the following characteristics.

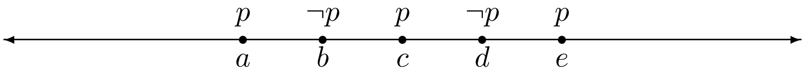
- $\mathcal{K}$ is a set of “worlds” (or “points”).
- R is a binary relation on $\mathcal{K}$. (Intuitively, if $H, H^{'} \in \mathcal{K}$, then $HRH^{'}$ means that $H^{'}$ is “reachable” (or “accessible”) from H).
- $\phi : \{\text{wff's}\} \times \mathcal{K} \to \{T, F\}$ such that

  1. For all worlds $H \in \mathcal{K}$, $\sigma(A) = \phi(A, H)$ is a propositional truth valuation.
  2. For all worlds $H \in \mathcal{K}$ and all wff's A, $\phi(\Box A, H) = T$ iff for all worlds $H^{'} \in \mathcal{K}$, $HRH^{'} \to \phi(A, H^{'})$.

  (Note that we could have defined ϕ in terms of propositional variables only, and then extended it to wff's.)

Proposition: ϕ is determined by its values on propositional variables; i.e., knowing $\phi(p_i, H)$ for all p_i and H determines ϕ for all formulas A. $\phi(A, H)$ can be determined by induction on the complexity of A.

Example: In tense logic, $\Box A$ means “A is true now and at all times in the future”. Pictorially, we might have the following “time line”:



At points (or worlds) a, c, and e, p is true, while at points b and d, p is false. So at all points at or before d, $\Box p$ is false. If at all points after d, p is true, then $\Box p$ is true at d. In tense logic, the relation R of a Kripke model is $\leq$. So all points in the future of a given point are reachable from the given point.

Definition: Let $\mathcal{M} = (\mathcal{K}, R, \phi)$. A is *true in* $\mathcal{M}$ *at* H, denoted by $(\mathcal{M}, H) \models A$, iff $\phi(A, H) = T$. A is *true in* $\mathcal{M}$, denoted by $\mathcal{M} \models A$, iff for all $H \in \mathcal{K}$, $(\mathcal{M}, H) \models A$. The pair $(\mathcal{K}, R)$ is called a *frame.* A is *valid on the frame* $(\mathcal{K}, R)$, denoted by $(\mathcal{K}, R) \models A$, iff for all Kripke models $\mathcal{M}$ of the form $(\mathcal{K}, R, -)$, $\mathcal{M} \models A$.

Example: In the above example, $(\mathcal{K}, R)$ is $(\mathbb{R}, \leq)$, and $\Box A \rightarrow A$ is valid on this frame. To see this, note that for all $H \in \mathbb{R}$, $\phi(\Box A \rightarrow A, H) = T$ iff $\phi(\Box A, H) = F$ or $\phi(A, H) = T$. Now suppose that $\phi(\Box A, H) = T$. Then $\phi(A, H^{'}) = T$ for all $H^{'}$ such that $H \leq H^{'}$. In particular, $\phi(A, H) = T$ since $\leq$ is reflexive. (Note that $\Box A \rightarrow A$ is not valid on $(\mathbb{R}, <)$ since $<$ is not reflexive. Also note that $\Box A \rightarrow \Box\Box A$ is valid on $(\mathbb{R}, \leq)$.)

Theorem: If $(\mathcal{K}, R)$ is a frame with reflexive R, then $\Box A \rightarrow A$ is valid on $(\mathcal{K}, R)$.

Theorem: If $(\mathcal{K}, R)$ is a frame with transitive R, then $\Box A \rightarrow \Box\Box A$ is valid on $(\mathcal{K}, R)$.

Theorem: If $(\mathcal{K}, R)$ is a frame on which $\Box A \rightarrow A$ is valid, then R is reflexive.

Proof: Suppose that R is not reflexive. Then there is an $H \in \mathcal{K}$ such that $\neg HRH$. Define ϕ by

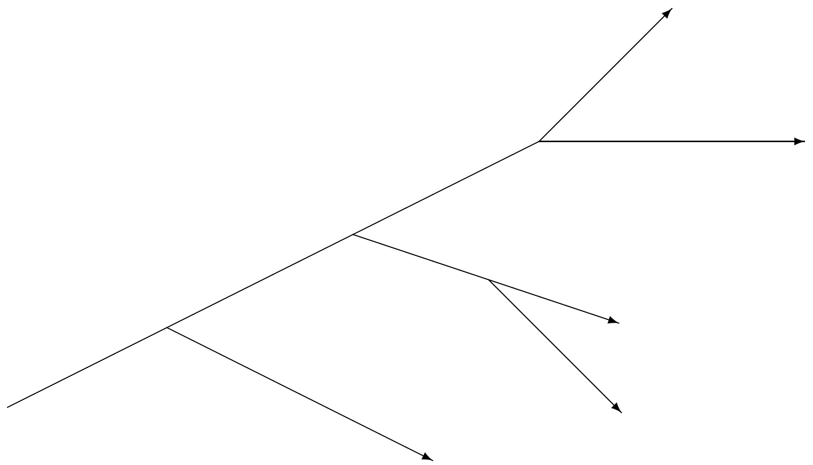
$$\begin{aligned} \phi(p, H) &= F \\ \phi(p, H^{'}) &= T, \text{ for all } H^{'} \neq H \\ \phi(q, -) &= \text{arbitrary for } q \neq p \end{aligned}$$

and extend ϕ in the unique way to get a truth valuation for all wff's. Now, $\phi(\Box A, H) = T$, since $\Box A$ is true in all worlds reachable from H and H is not reachable from itself. But $\phi(A, H) = F$ by definition, so $\phi(\Box A \rightarrow A, H) = F$. So we have constructed a model in which $\Box A \rightarrow A$ is not true. Hence $\Box A \rightarrow A$ is not valid which contradicts the hypothesis of the theorem. So R must be reflexive. □

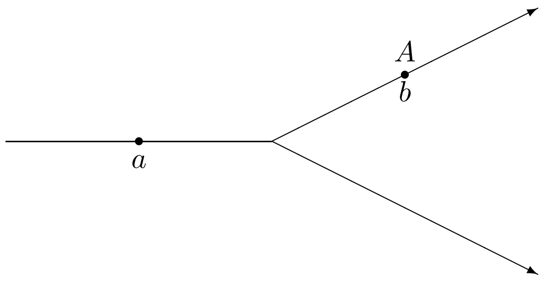
Up to now, we have considered the single modal operator □. But there may be more than one. Temporal logic uses two. F is used to represent "true at all times in the future", and G represents "true at all times in the past". P represents "true at some time in the future" (i.e., $\neg F \neg$), and H

represents “true at some time in the past” (i.e., $\neg G \neg$). In proofs of computer programs, each program P represents a modal operator $\Box_{\mathrm{P}}$. $A \rightarrow \Box_{\mathrm{P}} B$ can mean that if A is true before running P, then B is always true after running P.

The first example above portrayed what is called linear time. We could also have what is called branching time:



This is intended to represent the possibility of different futures for any particular point in time. In branching time models, formulas such as $\Box A \rightarrow A$ and $\Box A \rightarrow \Box\Box A$ are valid. As an example of a formula not valid in branching time models, take $\Diamond A \rightarrow \Box\Diamond A$ and consider the following model:



in which the formula A is true only at point b. Then at point a, $\Diamond A$ is true, but so is $\neg\Box\Diamond A$, since A is not true on the lower time line. So $\Diamond A \rightarrow \Box\Diamond A$ is not valid. (Note that it is not valid on $(\mathbb{R}, \leq)$ either. Also note that if every world is reachable from every other world, then $\Diamond A \to \Box\Diamond A$ is valid.)

Definition: A is *valid* iff A is valid on every frame.

Example: $\Box(A \to B) \to (\Box A \to \Box B)$ is valid for all formulas A and B.

Example: If A is valid, then $\Box A$ is valid since A being valid means that for all $(\mathcal{K}, R, \phi)$ and for all $H \in \mathcal{K}$, $\phi(A, H) = T$. Since this is true, then certainly for all $(\mathcal{K}, R, \phi)$ and for all $H, H^{'} \in \mathcal{K}$ such that $HRH^{'}$, $\phi(A, H^{'}) = T$. So $\Box A$ is valid.

Example: Letting $\bot$ ("false") be $p \wedge \neg p$, the formula $\neg\Box\bot$ is not valid. To show this, we have to find a model $\mathcal{M} = (\mathcal{K}, R, \phi)$ and an $H \in \mathcal{K}$ such that $(\mathcal{M}, H) \models \Box\bot$. Taking R to be empty gives such a model.

Proof Theory for Modal Logic

Let K be the set of modal formulas defined inductively by

Axiom 1. Every propositional tautology is in K.[1]

Axiom 2. $\Box(A \to B) \to (\Box A \to \Box B)$ is in K for all formulas A and B.

Rule 1. (Modus ponens.) If $A \in K$ and $A \to B \in K$, then $B \in K$ for all formulas A and B; i.e.,

$$\frac{A \quad A \to B}{B}.$$

Rule 2. (Necessitation.) If $A \in K$, then $\Box A \in K$; i.e.,

$$\frac{A}{\Box A}.$$

Theorem: (Soundness.) Every wff in K is valid.

Theorem: (Completeness.) Every valid formula is in K.

Proof: (of soundness.) Propositional tautologies are valid since ϕ respects the propositional connectives. Axiom 2 formulas are valid by a previous

[1] A propositional tautology is a valid formula in which all formulas of the form $\Box A$ are treated as propositional variables.

example. Modus ponens preserves validity for essentially the same reason that propositional tautologies are valid. I.e., if A and $A \to B$ are valid, then for all $\mathcal{M} = (\mathcal{K}, R, \phi)$ and $H \in \mathcal{K}$, $(\mathcal{K}, H) \models A$ and $(\mathcal{K}, H) \models A \to B$. So $(\mathcal{K}, H) \models B$ since $\phi(-, H)$ is a propositional truth valuation. Finally, the necessitation rule preserves validity by a previous example. So K is sound. $\Box$[2]

Definition: A theory (set of sentences) is *normal* iff it is closed under modus ponens and necessitation and contains $\mathcal{K}$.

In our discussion of theories, we will only talk about normal theories.

Notation: $K \vdash A$ means that $A \in K$.

For completeness, we will build a canonical model in which for every formula A, if $K \nvdash A$, then there is a point (a world) in the canonical model where A is false.

Definition: A propositional truth valuation ρ is *good* iff for all formulas A such that $K \vdash A$, $\rho(A) = T$.

In particular, if ρ is good, then $\rho(\Box A) = \rho(\Box(A \vee A))$ since

$$\begin{array}{ll} K \vdash A \to A \vee A & \text{tautology axiom} \\ K \vdash \Box(A \to A \vee A) & \text{by necessitation} \\ K \vdash \Box(A \to A \vee A) \to (\Box A \to \Box(A \vee A)) & \text{axiom 2.} \end{array}$$

So if ρ is good and $\rho(\Box A) = T$, then $\rho(\Box(A \vee A)) = T$. The converse is proved similarly.

Definition: R^+ is the binary relation on proposition truth valuations such that $\rho R^+ \rho^{'}$ iff for every wff A, $\rho(\Box A) = T$ implies that $\rho^{'}(A) = T$. This will force: "if ρ makes $\Box A$ true, then everything reachable from ρ makes A true."

Definition: The canonical structure is $(\mathcal{K}, R, \phi)$ where

$$\begin{aligned} \mathcal{K} &= \{\text{good propositional truth valuations}\}, \\ R &= R^+ \text{ restricted to } \mathcal{K}, \end{aligned}$$

[2] If $(\mathcal{K}, R)$ is a frame and A and $A \to B$ are valid on $(\mathcal{K}, R)$ then B is valid on $(\mathcal{K}, R)$. Likewise, if A is valid on $(\mathcal{K}, R)$ then $\Box A$ is valid on $(\mathcal{K}, R)$.

and ϕ is the unique truth valuation such that $\phi(p_i, \rho) = \rho(p_i)$ for all propositional variables p_i. What we will show next time is that ϕ makes $(\mathcal{K}, R, \phi)$ a Kripke model; i.e., $\phi(A, \rho) = \rho(A)$ for all $\rho \in \mathcal{K}$ and all wff's A.

Completeness (cont.), Compactness, Normal Theories

Math 260C - Mathematical Logic

June 2, 1989

Recall from last time, the theory K consisting of all propositional tautologies and formulas of the form $\Box(A \to B) \to (\Box A \to \Box B)$ and closed under modus ponens and necessitation. We showed the soundness of K (i.e., anything provable in K is valid in every frame), and started working on completeness (i.e., validity implies provability). The goal for completeness is, for A such that $K \not\vdash A$, to find an $\mathcal{M} = (\mathcal{K}, R, \phi)$ such that $(\mathcal{M}, H) \models \neg A$ for some $H \in \mathcal{K}$.

The $\mathcal{M}$ that we build will be a canonical model for K.[1] Recall that we started by defining *good* propositional truth valuations to be such that they assigned the value true to all formulas provable in K. Then we defined R^+ to be a relation on propositional truth valuations such that $\rho_1 R^+ \rho_2$ iff for all formulas A, $\rho_1(\Box A) = T$ implies that $\rho_2(A) = T$.

The $\mathcal{K}$ of our canonical model is then $\{\rho : \rho \text{ is good}\}$; i.e., the set of worlds is a set of propositional truth valuations. The R of our canonical model is the restriction of R^+ to $\mathcal{K}$; i.e., the restriction to good propositional truth valuations. This defines reachability to be such that if $\Box A$ is true in one world, then A is true in every other world reachable from the first world. Finally, we defined the ϕ of our canonical model to be such that $\phi(p_i, \rho) = \rho(p_i)$.

Now we have to show that ϕ satisifies the right conditions to make $(\mathcal{K}, R, \phi)$ a Kripke model. It does so iff it respects the propositional connectives and it makes A true in any world reachable from a world in which it makes $\Box A$ true; i.e., $\phi(\Box A, \rho) = T$ iff for all good $\rho^{'}$, $\rho R \rho^{'} \Rightarrow \phi(A, \rho^{'}) = T$. So our goal is to show that $\phi(A, \rho) = \rho(A)$. We'll do so with the following five claims.

[1] The construction of $\mathcal{M}$ works for any theory which contains K and is closed under modus ponens and necessitation.

Claim 1: If $K \nvdash A$, then there is a good propositional truth valuation ρ such that $\rho(A) = F$.

Proof: Since $K \nvdash A$, $K \cup \{\neg A\}$ is tautologically inconsistent.[2] So there is a propositional truth valuation ρ such that $\rho \models K \cup \{\neg A\}$[3] and ρ is clearly good. □

Claim 2: If $\phi(A, \rho) = \rho(A)$ for all wff's A and all good propositional truth valuations ρ, then K is complete.

Proof: Suppose that $K \nvdash A$. Then we want to show that A is not valid. So let ρ_A be the propositional truth valuation of claim 1. Then $\rho_A(A) = F$, and $\phi(A, \rho_A) = \rho_A(A) = F$. So A is not valid. □

Claim 3: If $\rho R^+ \rho^{'}$ and if ρ is good, then $\rho^{'}$ is good.

Proof: Suppose that $K \vdash A$. Then we want to show that $\rho^{'}(A) = T$. Now,

$$\begin{array}{rcll} K \vdash A & \Rightarrow & K \vdash \Box A & \text{by necessitation} \\ & \Rightarrow & \rho(\Box A) = T & \text{since } \rho \text{ is good} \\ & \Rightarrow & \rho^{'}(A) = T & \text{by defn. of } R^+. \ \Box \end{array}$$

Definition: The *box-theory* of a propositional truth valuation ρ, denoted by $\mathrm{Th}_\Box(\rho)$, is $\{B : \rho(\Box B) = T\}$.

Claim 4: If ρ is a good propositional truth valuation, then $\mathrm{Th}_\Box(\rho)$ is closed under tautological implication. (I.e., if $A \in \mathrm{Th}_\Box(\rho)$, and A tautologically implies C, then $C \in \mathrm{Th}_\Box(\rho)$.)

Proof: Suppose that $B_1, \ldots, B_k \in \mathrm{Th}_\Box(\rho)$ and that $\models B_1 \wedge \ldots \wedge B_k \rightarrow C$ (i.e., $B_1, \ldots, B_k$ tautologically imply C). Then we want to show that $C \in \mathrm{Th}_\Box(\rho)$. By definition of $\mathrm{Th}_\Box(\rho)$, $\rho(\Box B_i) = T$ for $1 \leq i \leq k$. So,

$$\begin{array}{ll} K \vdash B_1 \rightarrow (B_2 \rightarrow (\ldots (B_k \rightarrow C) \ldots)) & \text{tautology axiom} \\ K \vdash \Box(B_1 \rightarrow (B_2 \rightarrow (\ldots (B_k \rightarrow C) \ldots))) & \text{by necessitation} \\ K \vdash \Box B_1 \rightarrow \Box(B_2 \rightarrow (\ldots (B_k \rightarrow C) \ldots)) & \text{axiom 2} \\ \rho(\Box B_1 \rightarrow \Box(B_2 \rightarrow \ldots (B_k \rightarrow C) \ldots)) = T & \text{since } \rho \text{ is good} \\ \rho(\Box(B_2 \rightarrow (\ldots (B_k \rightarrow C) \ldots))) = T & \text{by defn. of } \rho. \end{array}$$

[2] $K \nvdash A$ implies that K does not tautologically imply A. Since K contains all tautologies and is closed under modus ponens, $K \cup \{\neg A\}$ has to be tautologically inconsistent. (Note that this reasoning relies on the compactness theorem for propositional logic.)

[3] I.e., for all $B \in K \cup \{\neg A\}$, $\rho(B) = T$.

Now do this $k-1$ more times to get $\rho(\Box C) = T$. So $C \in \mathrm{Th}_\Box(\rho)$. □

Claim 5: For all wff's A and all good propositional truth valuations ρ, $\phi(A, \rho) = \rho(A)$.

Proof: By induction on the complexity of A.

Case 1. A is p_i. Then the claim is true by definition of ϕ.

Case 2. A is $B \wedge C$, $B \vee C$, $\neg B$, or $B \rightarrow C$. Then the claim is true by induction and since both ρ and ϕ respect propositional connectives.

Case 3. A is $\Box B$. Then, by definition of ϕ, $\phi(\Box B, \rho) = T$ iff for all good $\rho^{'}$, $\rho R \rho^{'} \Rightarrow \phi(\rho^{'}, B) = T$. Or, by induction, $\phi(\Box B, \rho) = T$ iff for all good $\rho^{'}$, $\rho R \rho^{'} \Rightarrow \rho^{'}(B) = T$. So we want to show that

$$\rho(\Box B) \Leftrightarrow \text{for all good } \rho^{'},\ \rho R \rho^{'} \Rightarrow \rho^{'}(B) = T.$$

$\Rightarrow$:[4] If $\rho(\Box B) = T$ and if $\rho R \rho^{'}$, then $\rho^{'}(B) = T$ by definition of R.

$\Leftarrow$: Here, we want to show that if $\rho(\Box B) = F$, then there is a good $\rho^{'}$ such that $\rho R \rho^{'}$ and $\rho^{'}(B) = F$. So suppose that $\rho(\Box B) = F$. Then $B \notin \mathrm{Th}_\Box(\rho)$ by definition of $\mathrm{Th}_\Box(\rho)$. So $\mathrm{Th}_\Box(\rho) \cup \{\neg B\}$ is tautologically inconsistent by claim 4. So there is a propositional truth valuation $\rho^{'}$ such that $\rho^{'} \models \mathrm{Th}_\Box(\rho) \cup \{\neg B\}$. Hence $\rho R^{+} \rho^{'}$ since $\rho^{'} \models \mathrm{Th}_\Box(\rho)$. And $\rho^{'}$ is good by claim 3. So we have found a good $\rho^{'}$ such that $\rho R \rho^{'}$ and $\rho^{'}(B) = F$. □

The proof of the completeness theorem for K is now immediate by claims 5 and 2. Claim 5 shows that if $K \not\vdash A$, then there is a world in the canonical model where A is false. And claim 2 shows that it suffices to establish claim 5 in order to show that K is complete.

Compactness Theorem: If Γ is a set of formulas and for every finite subset $A_1, \ldots, A_k$ of Γ there is a Kripke model $\mathcal{M}$ and a world $H \in \mathcal{M}$ such that $(\mathcal{M}, H) \models A_1 \wedge \ldots \wedge A_k$, there there is a world ρ in the canonical model such that for all $A \in \Gamma$, $\rho(A) = T$.

[4] Note that we have defined R in order to make this direction work.

Proof: By the compactness theorem for propositional logic, there is a truth valuation ρ such that $\rho \models K \cup \Gamma$ since every subset of Γ is tautologically consistent with K.[5] $\Box$

Normal Theories

Definition: S is a *normal theory* if $S \supseteq K$ and is closed under modus ponens and necessitation.

Definition: [6]

$$\begin{array}{rll} T & \text{is} & K \cup \{\Box A \to A\}, \\ S4 & \text{is} & T \cup \{\Box A \to \Box\Box A\}, \\ K4 & \text{is} & K \cup \{\Box A \to \Box\Box A\}, \\ S5 & \text{is} & T \cup \{\Diamond A \to \Box\Diamond A\}, \\ G & \text{is} & K \cup \{\Box(\Box A \to A) \to \Box A\}, \\ B & \text{is} & K \cup \{A \to \Box\Diamond A\}, \text{ and} \\ TB & \text{is} & T \cup \{A \to \Box\Diamond A\}. \end{array}$$

Each theory above constrains the relation R that can be used in a model for that theory. The additional axiom for T expresses reflexivity. $S4$ expresses reflexivity and transitivity. The $K4$-axiom expresses transitivity. $S5$ expresses equivalence relations. The G-axiom is the Gödel-Löb axiom; it expresses transitivity and no sequence of the form $\rho_1 R \rho_2$, $\rho_2 R \rho_3$, $\rho_3 R \rho_4$, $\ldots$; i.e., R^{-1} is well-founded. In particular, R is not reflexive. The B-axiom expresses symmetry. And TB expresses reflexivity and symmetry.

These theories are not independent. For example, the $K4$-axiom is in G. It is hard to prove this, but two other examples are embodied in the following theorems.

[5] I.e., every subet of Γ is true in some Kripke model, and every Kripke model contains K (i.e., every consequence of K is valid in any Kripke model).

[6] *Motivation:* If we want $\Box A$ to mean that A is true now and at all times in the future, then we would want a theory like T. To show that T is consistent, we would have to show that $\Box A \to A$ is consistent with K; i.e., find a Kripke model and a world in which $K \cup \{\Box A \to A\}$ is true.

Theorem: $S5 \vdash A \rightarrow \Box\Diamond A$ (the B-axiom).

Proof: Since $A \rightarrow \Diamond A$ is the contrapositive of $\Box\neg A \rightarrow \neg A$, and since $S5 \supset T$ (i.e., $S5$ contains $\Box\neg A \rightarrow \neg A$ as an axiom),

$$\begin{array}{lll} S5 \vdash A \rightarrow \Diamond A & & (1) \\ S5 \vdash A \rightarrow \Box\Diamond A & \text{by } S5\text{-axiom. } \Box & (2) \end{array}$$

Theorem: $S5 \vdash \Box A \rightarrow \Box\Box A$ (the $K4$-axiom).

Proof:

$$\begin{array}{lll} S5 \vdash \Diamond\Box A \rightarrow \Box A & \text{by } S5\text{-axiom} & (3) \\ S5 \vdash \Box(\Diamond\Box A \rightarrow \Box A) & \text{by necessitation} & (4) \\ S5 \vdash \Box\Diamond\Box A \rightarrow \Box\Box A) & \text{by axiom 2} & (5) \\ S5 \vdash \Box A \rightarrow \Diamond\Box A) & \text{by (1)} & (6) \\ S5 \vdash \Diamond\Box A \rightarrow \Box\Diamond\Box A) & \text{by } S5\text{-axiom} & (7) \\ S5 \vdash \Box A \rightarrow \Box\Box A) & \text{by (6), (7), and (4). } \Box & (8) \end{array}$$

Definition: The *correspondence properties* for the following theories are defined by the following table:

theory	correspondence property
T	reflexive
$K4$	transitive
$S4$	reflexive and transitive
$S5$	equivalence
G	transitive and R^{-1} is a well-founded strict partial order
B	symmetric
TB	reflexive and symmetric

Soundess and Completeness Theorems: A wff A is a consequence of the theory S iff A is valid on any frame in which R satisfies the correspondence property for S.

E.g., a formula true in any transitive frame is in $K4$.

Soundness and Completeness for Normal Theories

Math 260C - Mathematical Logic

June 5, 1989

Recall from last time, the following normal theories and their correspondence properties:

$$\begin{array}{lcll}
T & \equiv & K \cup \{\Box A \to A\} & \text{reflexive} \\
S4 & \equiv & T \cup \{\Box A \to \Box\Box A\} & \text{transitive, reflevive} \\
K4 & \equiv & K \cup \{\Box A \to \Box\Box A\} & \text{transitive} \\
S5 & \equiv & T \cup \{\Diamond A \to \Box\Diamond A\} & \text{equivalence relation} \\
G & \equiv & K \cup \{\Box(\Box A \to A) \to \Box A\} & \text{transitive, } R^{-1} \text{ well-founded} \\
B & \equiv & K \cup \{A \to \Box\Diamond A\} & \text{symmetric}
\end{array}$$

Let S be any of the above normal theories.

Soundness Theorem: Any consequence of (member of) S is valid on any frame with the correspondence property of S.

Proof:

> *Lemma:* The set of formulas valid on a frame $(\mathcal{K}, R)$, $\{A : (\mathcal{K}, R) \models A\}$, is closed under modus ponens and necessitation.
>
> *Proof*: By induction on the complexity of A.[1]

By the lemma, we just need to verify soundness for the axioms. In particular, we just need to check the reflexive axiom, the symmetric axiom, the transitive axiom, and the transitive plus R^{-1} well-founded axiom since the other axioms are implied by these. I.e., we need to show that if A is any wff, then

[1] For modus ponens, if A and $A \to C$ are valid, then C is also valid, since any propositional truth valuation respects the propositional connectives. Similarly, for necessitation, if A is valid, then $\Box A$ is also valid, since A is true in any world reachable from one in which A is true.

a) if R is reflexive, then $(\mathcal{K}, R) \models \Box A \rightarrow A$,

b) if R is symmetric, then $(\mathcal{K}, R) \models A \rightarrow \Box\Diamond A$,

c) if R is transitive, then $(\mathcal{K}, R) \models \Box A \rightarrow \Box\Box A$, and

d) if R is transitive and has no infinite sequence of related worlds $H_1 R H_2$, $H_2 R H_3$, …, then $(\mathcal{K}, R) \models \Box(\Box A \rightarrow A) \rightarrow \Box A$.

Cases (a) and (c) were done earlier[2] So all we need to verify are cases (b) and (d).

b) Suppose that R is symmetric, and take any $H \in \mathcal{K}$ such that $\phi(A, H) = T$. Then, for all $H^{'}$, $HRH^{'} \rightarrow \phi(\Diamond A, H^{'}) = T$, since $H^{'}RH$ by the supposed symmetry of R.

d) Suppose that R is transitive and that R^{-1} is well-founded.[3] Further suppose, towards a contradiction, that there is an $H \in \mathcal{K}$, a wff A, and a propositional truth valuation ϕ on $(\mathcal{K}, R)$ such that $\mathcal{M} = (\mathcal{K}, R, \phi)$ is a Kripke model and that

$$(\mathcal{M}, H) \models \Box(\Box A \rightarrow A) \wedge \neg\Box A.$$

Let

$$Y = \{H^{'} : HRH^{'} \text{ and } (\mathcal{M}, H) \models \neg A\},$$

and let H_0 be maximal in Y (i.e., forall $H^{'} \in Y$, $\neg H_0 R H^{'}$). Note that Y is not empty since $(\mathcal{M}, H) \models \neg\Box A$ and that H_0 exists since R^{-1} is assumed to be well-founded.

Claim: $(\mathcal{M}, H) \models \Box A$.

Proof: If not, then there is an $H^{'}$ such that $H_0 R H^{'}$ and $(\mathcal{M}, H^{'}) \models \neg A$. But then, since R is assumed to be transitive, $HRH^{'}$. So $H^{'} \in Y$. But this contradicts our choice of H_0 as being maximal. □

[2] See examples, 5/26.

[3] An equivalent condition for the well-foundedness of R^{-1} is that for any subset $Y \subseteq \mathcal{K}$, there is a "maximal" $H \in Y$ such that for all $H^{'} \in Y$, $\neg HRH^{'}$.

So $(\mathcal{M}, H) \models \Box(\Box A \to A)$ implies that $(\mathcal{M}, H_0) \models \Box A \to A$ since HRH_0. Thus, by the claim, $(\mathcal{M}, H_0) \models A$. But this contradicts $H_0 \in Y$ by the definition of Y. $\Box$

Completeness Theorem: Let A be a wff. If A is valid on every frame satisfying the correspondence property for S, then $S \vdash A$.

Proof: (Except for G.) As in the proof of soundness for K, we'll suppose that $S \not\vdash A$ and then find a frame and a world that doesn't satisfy the correspondence property. Similar to the earlier canonical model construction, let $\mathcal{K} = \{\rho : \rho \text{ is } S\text{-good}\}$, let R be the restriction of R^+ to $\mathcal{K}$, and let $\phi(A, \rho) = \rho(A)$. Then claims 1, 3, 4, and 5 of the proof of the soundness of K apply here with S-good replacing good. What we get is a canonical model $\mathcal{M} = (\mathcal{K}, R, \phi)$ for S such that if $S \not\vdash A$, then there is an $H \in \mathcal{K}$ with $(\mathcal{M}, H) \models \neg A$. It remains to show that $(\mathcal{K}, R)$ satisfies the right correspondence property for S. Since we're not proving the completeness for G, we only have to check reflexivity, symmetry, and transitivity, since $S4$ and $S5$ can be expressed in terms of these properties.[4] I.e., we need to show that for all formulas A,

a) if $\Box A \to A$ is in S, then R is reflexive,

b) if $\Box A \to \Box\Box A$ is in S, then R is transitive, and

c) if $A \to \Box\Diamond A$ is in S, then R is symmetric.

So,

a) Let ρ be S-good. Then we want to show that $\rho R \rho$; i.e., for all formulas A, if $\rho(\Box A)$ then $\rho(A) = T$. But this is obvious because $\rho(\Box A \to A) = T$.

b) Let ρ_1, ρ_2, and ρ_3 be S-good, and suppose that $\rho_1 R \rho_2$ and $\rho_2 R \rho_3$. Then we want to show that $\rho_1 R \rho_3$. Let A be any wff. Then

$$\begin{array}{lcll} \rho_1(\Box A) = T & \Rightarrow & \rho_1(\Box\Box A) = T & \text{since } \rho_1(\Box A \to \Box\Box A) = T \\ & \Rightarrow & \rho_2(\Box A) = T & \text{since } \rho_1 R \rho_2 \\ & \Rightarrow & \rho_3(A) = T & \text{since } \rho_2 R \rho_3. \end{array}$$

[4] Recall that $S5 \vdash \Box A \to \Box\Box A$.

c) Let ρ_1 and ρ_2 be S-good, and suppose that $\rho_1 R \rho_2$. Then we want to show that $\rho_2 R \rho_1$. Suppose, towards a contradiction, that $\rho_2(\Box A) = T$ and that $\rho_1(A) = F$ (i.e., $\neg \rho_2 R \rho_1$). Then, since $\rho_1(\neg A) = T$, $\rho_1(\Box \Diamond \neg A = T)$ by the hypothesis applied to $\neg A$. And since $\rho_1 R \rho_2$, $\rho_2(\Diamond \neg A) = T$; i.e., $\rho_2(\neg \Box A) = T$. But this contradicts $\rho_2(\Box A) = T$. $\Box$

The reason that the above construction doesn't work for G is essentially because we don't get well-founded models. Next time, to prove completeness for G, we'll find a finite frame on which A is false if $G \not\vdash A$. Finite frames are always well-founded provided that they are not reflexive.

Completeness via the Filtration Method, Arithmetical Completeness of G

Math 260C - Mathematical Logic

June 9, 1989

Last time, we proved the completeness of several normal theories by constructing canonical models but were unable to apply the construction process to G. Today, we'll re-prove the completeness theorems and get a stronger result.

Completeness Theorem: Let S be K, T, K4, S4, S5, B, TB, or G. If $S \nvdash A$, then there is a Kripke model $\mathcal{M} = (\mathcal{K}, R, \phi)$ such that

1. $\mathcal{K}$ is finite,
2. $(\mathcal{K}, R)$ satisfies the correspondence property of S, and
3. there is an $H \in \mathcal{K}$ such that $(\mathcal{M}, H) \models \neg A$.

Proof: Later.

$\mathcal{K}$ being finite is the key here. By the construction process, it will turn out that $|\mathcal{K}| \leq 2^{|A|}$. This gives us the following corollary.

Corollary: S is decidable.

In fact, there is an exponential space algorithm (double exponential time algorithm) for deciding membership in S.[1] We just look at all finite models whose size is $\leq 2^{|A|}$. If any of them satisfy $\neg A$, then S $\nvdash A$; otherwise, S $\vdash A$.

[1] Even if we didn't have the bound $|\mathcal{K}| \leq 2^{|A|}$, the corollary would still hold. We could interleave the enumeration of all finite models and all consequences of A under modus ponens and necessitation until we find either a proof of A or a Kripke model in which A is false at some world.

The Kripke model construction process for proving the completeness of the various theories is based on the *filtration method.* The idea is to take a finite "homomorphic image" of the canonical model for S. From now on, for notational convenience, let $\mathcal{M} = (\mathcal{K}, R, \phi)$ denote the canonical model for S.[2] $\mathcal{K}$ is big because we have propositional truth valuations over an infinite domain. But in order to decide whether A is satisfied by a given model, we only need propositional truth valuations over A and its subformulas. More specifically then, the idea behind the filtration method is to identify $\rho, \rho' \in \mathcal{K}$ whenever ρ and ρ' agree on subformulas of A; what ρ and ρ' do on other formulas is immaterial to the satisfiability of A.

Definition: B is a *modal subformula* of A iff either

1. B is A,
2. A is $A_1 \wedge A_2$ or $A_1 \vee A_2$ or $A_1 \rightarrow A_2$ and B is a modal subformula of A_1 or A_2, or
3. A is $\neg A_1$ or $\Box A_1$ and B is a modal subformula of A_1.

Now, let's fix A such that $S \nvdash A$. Also, let Σ be $\{B : B$ is a modal subformula of $A\}$.

Definition: If ρ and ρ' are propositional truth valuations, then ρ and ρ' are *equivalent*, denoted by $\rho \sim \rho'$, iff for every $B \in \Sigma$, $\rho(B) = \rho'(B)$. The *equivalence class* of ρ, denoted by $[\rho]$, is $\{\rho' : \rho \sim \rho'\}$.[3]

Now we're ready to prove the completeness theorem. Let $\mathcal{K}' = \{[\rho] : \rho$ is S-good$\}$. This is the set of worlds in our finite Kripke model.[4] The proof of completeness will be broken down into four cases for the theories K, T, K4, S4, S5, B, TB, and G. In each case, we'll first define a reachability relation

[2] If S is G, then $(\mathcal{K}, R, \phi)$ is still a Kripke model, but R^{-1} may not be well-founded since we haven't already proved the completeness of G. We'll take care of this problem later.

[3] Note that there are less than $2^{|A|}$ equivalence classes since there are less than $|A|$ subformulas of A.

[4] One obvious set of worlds to try would be the set of all worlds reachable from a world in which $\neg A$ is true. This won't work since it doesn't guarantee a finite model. Note that we're not building as rich a model as in the earlier completeness theorem. In that theorem, compactness was built into the model; i.e., if every finite subset of an infinite set of formulas had a model, then the whole set had a model. In the current situation, we don't get compactness.

tailored for the particular correspondence properties and a truth valuation in terms of that reachability relation. Then we'll show that $\mathcal{K}'$ and the new reachability relation and truth valuation is the desired finite Kripke model.

Case 1. S is K, T, B, or TB; i.e., R may be reflexive and/or symmetric but is not transitive.

Definition: The new reachability relation, R^σ, is defined as $[\rho]R^\sigma[\rho']$ iff there is a $\rho_1 \in [\rho]$ and a $\rho_2 \in [\rho']$ such that $\rho_1 R \rho_2$.

Definition: The new truth valuation ϕ' is defined on propositional variables as

$$\phi'(p_i, [\rho]) = \left\{ \begin{array}{ll} \rho(p_i) & \text{if } p_i \in \Sigma \\ F & \text{if } p_i \notin \Sigma \end{array} \right.$$

and $\phi'(B, [\rho])$ is the unique value compatible with the definitions of R^σ and ϕ' on propositional variables.

Now we have to show that $(\mathcal{K}', R^\sigma, \phi')$ is the desired Kripke model.

Lemma 0: For some $[\rho] \in \mathcal{K}'$, $\rho(A) = F$.

Proof: Take $\rho \in \mathcal{K}$ such that $\rho(A) = F$. Such a ρ exists by the construction in the earlier completeness theorem. (Note that this works even for G.) □

Lemma 1: If $\rho R \rho'$, then $[\rho]R^\sigma[\rho']$.

Proof: Trivial by the definition of R^σ. □

Lemma 2: For all $B \in \Sigma$ and all S-good ρ, $\phi'(B, [\rho]) = \rho(B)$.

Proof: By induction on the complexity of B.

Basis: B is a propositional variable. Then the lemma is true by the definition of ϕ'.

Induction: B is $\neg C$, $C_1 \wedge C_2$, $C_1 \vee C_2$, or $C_1 \rightarrow C_2$ for $C, C_1, C_2 \in \Sigma$. Then the lemma is true by induction since ϕ' respects the propositional connectives.

B is $\Box C$ for $C \in \Sigma$. We want to show that $\phi'(\Box C, [\rho]) = \rho(\Box C)$.

If $\rho(\Box C) = T$, then for all ρ' such that $\rho R \rho'$, $\rho'(C) = T$. So for all ρ' such that $[\rho] R^\sigma [\rho']$, $\rho'(C) = T$. Hence $\phi'(C, [\rho']) = T$ by the induction hypothesis. So $\phi'(\Box C, [\rho']) = T$ by the definition of ϕ'.

If $\rho(\Box C) = F$, then, from the proof of the earlier completeness theorem, there is a $\rho' \in \mathcal{K}$ such that $\rho'(C) = F$ and $\rho R \rho'$. Hence, by lemma 1, $[\rho] R^\sigma [\rho']$. So $\phi'(C, [\rho']) = F$ by the induction hypothesis and since $[\rho] R^\sigma [\rho']$ and $\rho'(C) = F$ (i.e., there is a reachable world ρ' at which C is false). So $\phi'(\Box C, [\rho']) = F$ by the definition of ϕ'. □

This proves the completeness theorem for S = K since we've built a finite Kripke model in which A is false at a world.

Lemma 3: If R is reflexive, then so is R^σ. And if R is symmetric, then so is R^σ.

Proof: Easy using the definition of R^σ. □

This proves the completeness theorem for S = T, B, or TB.

Case 2. S is K4 or S4; i.e., S is transitive and possibly reflexive, but not symmetric.[5] We'll use the same set of worlds, $\mathcal{K}'$, but we need a different reachability relation and truth valuation.

Definition: $[\rho] R^\tau [\rho']$ iff for all $\Box B \in \Sigma$,

$$\rho(\Box B) = T \quad \Rightarrow \quad \rho'(B) = T \text{ and } \rho'(\Box B) = T.$$

Definition: The *box-sigma-theory* of a propositional truth valuation ρ, denoted by $\mathrm{Th}_{\Box\Sigma}$, is $\{B : \rho(\Box B) = T \text{ and } \Box B \in \Sigma\}$.

[5] The reason that we need another case for transitivity is that it is harder to handle. To see why, take the reachability relation R^σ from case 1, and suppose that $[\rho_1] R^\sigma [\rho_2]$ and $[\rho_2] R^\sigma [\rho_3]$. This would imply that $\rho'_1 R \rho'_2$ and $\rho''_2 R \rho'_3$ for some $\rho'_1 \in [\rho_1]$, $\rho'_2, \rho''_2 \in [\rho_2]$, and $\rho'_3 \in [\rho_3]$, where ρ'_2 may not be the same as ρ''_2. So the transitivity of R does not necessarily imply the transitivity of R^σ.

Using the definition of $\mathrm{Th}_{\Box\Sigma}$, an equivalent definition for R^τ is $[\rho]R^\tau[\rho']$ iff $\mathrm{Th}_{\Box\Sigma}(\rho) \subseteq \mathrm{Th}_{\Box\Sigma}(\rho')$ and for all $\Box B \in \Sigma$, $\rho(\Box B) = T$ implies that $\rho'(B) = T$.

Definition: The new truth valuation ϕ' is defined as in case 1 except in terms of R^τ instead of R^σ.

Lemma 4: R^τ is transitive.

Proof: Obvious from the original definition of R^τ.[6]

Lemma 5: If $\rho R \rho'$, then $[\rho]R^\tau[\rho']$.

Proof: We want to show that if $\rho R \rho'$ and $\rho(\Box B) = T$, then $\rho'(\Box B) = T$, and $\rho'(B) = T$. By definition of R, $\rho(\Box B) = T$ implies that $\rho'(B) = T$. And $\rho(\Box B) = T$ implies that $\rho(\Box\Box B) = T$ since ρ is S-good and $\mathrm{S} \vdash \Box B \to \Box\Box B$. So $\rho'(\Box B) = T$ by the definition of R. □

Lemma 6: For all $B \in \Sigma$ and all S-good ρ, $\phi'(B, [\rho]) = \rho(B)$.

Proof: By induction on the complexity of B. (Very similar to the proof of lemma 2. Just replace R^σ by R^τ and usages of lemma 1 by usages of lemma 5.) □

This proves the completeness theorem for S = K4 since if $\mathrm{K4} \nvdash A$, we have a finite Kripke model and a world in which A is false.

Lemma 7: If R is reflexive (i.e., S = S4), then so is R^τ.

Proof: Obvious using the definition of R^τ and since R is reflexive. □

This proves the completeness theorem for S = S4.

[6] I.e., if $[\rho_1]R^\tau[\rho_2]$ and $[\rho_2]R^\tau[\rho_3]$, then there are $\rho_1' \in [\rho_1]$, $\rho_2' \in [\rho_2]$, and $\rho_3' \in [\rho_3]$ such that

$$\begin{aligned} \rho_1'(\Box B) = T &\Rightarrow \rho_2'(\Box B) = T \\ &\Rightarrow \rho_3'(\Box B) = T \text{ and } \rho_3'(B) = T. \end{aligned}$$

Case 3. S is S5; i.e., S has the equivalence relation property. This case is a combination of cases 1 and 2 and won't be proved in detail. The idea is to define the reachablility relation R^ϵ by $[\rho]R^\epsilon[\rho']$ iff $\mathrm{Th}_{\Box\Sigma}(\rho) = \mathrm{Th}_{\Box\Sigma}(\rho')$ and for all $\Box B \in \Sigma$ $\rho(\Box B) = T$ implies that $\rho'(B) = T$.

Then, transitivity and reflexivity are shown exactly as before, and symmetry is easily proved.[7]

The counterpart to lemma 5 is

Lemma 8: If $\rho R \rho'$, then $[\rho]R^\epsilon[\rho']$.

Proof: Use the fact that R^ϵ is an equivalence relation. □

Case 4. S = G.[8]

Definition: $[\rho]R^\gamma[\rho']$ iff $\mathrm{Th}_{\Box\Sigma}(\rho) \subsetneq \mathrm{Th}_{\Box\Sigma}(\rho')$ and for all $\Box B \in \Sigma$, $\rho(\Box B) = T$ implies that $\rho'(B) = T$.

Note that this is almost like the definition of R^τ except for the *proper* inclusion of $\mathrm{Th}_{\Box\Sigma}(\rho)$. This is necessary to force irreflexivity.

Definition: The new truth valuation ϕ' is defined as before except in terms of R^γ.

Lemma 9: R^γ is transitive and irreflexive.

Proof: Transitivity is as before (in lemma 4). Irreflexivity is obvious from the definition of R^γ (i.e., $[\rho]R^\gamma[\rho]$ is impossible).

Lemma 10: For all $B \in \Sigma$ and all G-good ρ, $\phi'(B, [\rho]) = \rho(B)$.

Proof: By induction on the complexity of B. As in lemmas 2 and 6, the proof is easy for the cases where B is either a propositional variable or composed by propositional connectives from formulas in Σ. The case where B is $\Box C$ is a little different.

[7] Or, more simply, define $[\rho]R^\epsilon[\rho']$ iff for all $\Box B \in \Sigma$ $\mathrm{Th}_{\Box\Sigma}(\rho) = \mathrm{Th}_{\Box\Sigma}(\rho')$. Then, transitivity, reflexivity, and symmetry are easily shown.

[8] Recall the G-axioms: $\Box(\Box A \rightarrow A) \rightarrow \Box A$, and $\Box A \rightarrow \Box\Box A$. That the second one follows from the first is shown in a footnote on page 268 of Boolos and Jeffrey. Also, the proof for case 4 above is similar to a proof by Robert Solovay in "Provability Interpretations of Modal Logic", *Israel Journal of Mathematics*, 25(1976), pp. 287-304.

If $\rho(\Box C) = T$, then for all ρ' such that $[\rho] R^\gamma [\rho']$, $\rho'(C) = T$ by definition of R^γ. So for all ρ' such that $[\rho] R^\gamma [\rho']$, $\phi'(C, \rho') = T$ by the induction hypothesis. Hence $\phi'(\Box C, \rho') = T$ by the definition of ϕ'.

If $\rho(\Box C) = F$, then we want to show that $\phi'(\Box C, \rho) = F$; i.e., we want a ρ' such that $[\rho] R^\gamma [\rho']$ and $\phi(C, \rho') = F$. By the induction hypothesis, it suffices to find a ρ' such that $[\rho] R^\gamma [\rho']$ and $\rho'(C) = F$. More specifically, if $\mathrm{Th}_{\Box\Sigma}(\rho) = \{D_1, \ldots, D_k\}$, then we want a ρ' such that

1. ρ' is G-good,
2. $\rho'(C) = F$,
3. $\rho'(\Box D_i) = T$ for $1 \leq i \leq k$,
4. $\rho'(\Box C) = T$, and
5. $\rho'(D_i) = T$, for $1 \leq i \leq k$.[9]

Let $D = D_1 \wedge \ldots \wedge D_k$, and note that $\Box D \rightarrow \Box D_1 \wedge \ldots \wedge \Box D_k$ is a consequence of K. If we can show that $\mathrm{G} \cup \{\neg C, \Box C, \Box D, D\}$ is tautologically consistent, then by compactness[10], there will be a ρ' such that conditions 1-5 hold.

Suppose, for the sake of contradiction, that $\mathrm{G} \cup \{\neg C, \Box C, \Box D, D\}$ is not tautologically consistent. Then

$$\begin{array}{llll}
\mathrm{G} & \vdash & C \vee \neg\Box C \vee \neg\Box D \vee \neg D & & (1) \\
& & \Box D \wedge D \rightarrow (\Box C \rightarrow C) & & (2) \\
& & \Box(\Box D \wedge D) \rightarrow \Box(\Box C \rightarrow C) & \text{by necessitation and the} & \\
& & & \text{axiom } \Box(X \rightarrow Y) \rightarrow \Box X \rightarrow \Box Y & (3) \\
& & \Box(\Box C \rightarrow C) \rightarrow \Box C) & \text{by the Gödel-Löb axiom} & (4) \\
& & \Box\Box D \wedge \Box D \rightarrow \Box(\Box D \wedge D) & \text{a consequence of K} & (5) \\
& & \Box D \rightarrow \Box\Box D & \text{by G-axiom} & (6) \\
& & \Box D \rightarrow \Box C & \text{by (3)-(6).} & (7)
\end{array}$$

In particular, $\mathrm{G} \vdash \Box D_1 \wedge \ldots \wedge \Box D_k \rightarrow \Box C$. But this contradicts the facts that $\rho(\Box D_i) = T$ for $1 \leq i \leq k$, that $\rho(\Box C) = F$, and that ρ is G-good. So $\mathrm{G} \cup \{\neg C, \Box C, \Box D, D\}$ must be tautologically consistent, and hence there is a ρ' satisfying conditions 1-5. □

[9] Conditions 3, 4, and 5 force $\mathrm{Th}_{\Box\Sigma}(\rho) \subsetneq \mathrm{Th}_{\Box\Sigma}(\rho')$ since we're assuming that $\Box C \notin \mathrm{Th}_{\Box\Sigma}(\rho)$.

[10] Remember, G is infinite.

So the completeness of G is established, and the entire completeness theorem is proved.

Arithmetical Completeness of G

Definition: Let Ψ be a mapping from propositional variables to sentences in the language of PA, and extend Ψ to all wff's A such that

1. Ψ respects propositional connectives, and
2. $\Psi(\Box A) = \mathrm{Thm}_{\mathrm{PA}}(\underline{\ulcorner \Psi(A) \urcorner})$.

So we interpret $\Box A$ as "A is PA-provable".

Arithmetical Completeness and Soundness Theorem: Let A be a wff. Then $\mathrm{G} \vdash A$ iff for all Ψ, $\mathrm{PA} \vdash \Psi(A)$.[11]

Recall Löb's theorem: if

$$\mathrm{PA} \vdash \mathrm{Thm}_{\mathrm{PA}}(\underline{\ulcorner \mathrm{Thm}_{\mathrm{PA}}(\underline{\ulcorner A \urcorner}) \to A \urcorner}),$$

then $\mathrm{PA} \vdash \mathrm{Thm}_{\mathrm{PA}}(\underline{\ulcorner A \urcorner})$. The arithmetical completeness and soundness theorem generalizes this by allowing arbitrarily nested $\Box$ formulas. In particular, $\Box(\Box p_i \to p_i) \to \Box p_i)$ iff

$$\mathrm{PA} \vdash \mathrm{Thm}_{\mathrm{PA}}(\underline{\ulcorner \mathrm{Thm}_{\mathrm{PA}}(\underline{\ulcorner A \urcorner}) \to A \urcorner}) \to \mathrm{Thm}_{\mathrm{PA}}(\underline{\ulcorner A \urcorner})$$

for all sentences A.

[11] Boolos and Jeffrey prove the soundness part of this theorem.