References:
Hartley Rogers, Theory of recursive functions
Robert Soare, Recursively enumerable sets and degrees
Barry Cooper, Computability theory

UR-Basic Programming Language

Variables are any string of letters or numerals, A-Za-z0-9.

Statements are of the form
Let \( X = X + 1 \)
Let \( X = X - 1 \)
If \( X \leq Y \) then goto \( k \)

where \( X \) and \( Y \) are any variables and \( k \) is a nonnegative integer, i.e. \( k \in \omega \).

A UR-Basic program is a sequence \( S_0, S_1, S_2, \ldots, S_n \) of statements. Variables only take on nonnegative integer values. The symbol \( - \) means subtraction unless the result is negative and then it yields zero. The program halts if we go to any line \( k > n \).

A function \( f : \omega \to \omega \) is UR-Basic computable iff there exists a program \( P \), designated input variable \( X \) and output variable \( Y \) such that for any \( n \in \omega \) if we put \( X = n \) and all other variables zero and start with the first statement of \( P \), then \( P \) eventually halts with \( f(n) \) in variable \( Y \). There is a similar definition for \( f : \omega^m \to \omega \) to be UR-Basic computable.

Basic:

Go to k
Continue
Let Y=X

UR-Basic:

If \( X \leq X \) then goto \( k \)
Let Donothing=Donothing+1
Let Y=X
1 If \( X \leq Y \) then go to 4
2 Let Y=Y+1
3 Go to 1
4 If \( Y \leq X \) then go to 7
5 Let \( Y = Y - 1 \)
6 Go to 4
7 Continue

Constants
0 this is a variable - we agree never to change it
1 \hspace{1cm} \text{let } 1 = 1 + 1

2 \hspace{1cm} \text{Let } 2 = 2 + 1
\hspace{1cm} \text{Let } 2 = 2 + 1

\text{If } X < Y \text{ then goto } k
\hspace{1cm} \text{Let } \text{temp}X = X
\hspace{1cm} \text{Let } \text{temp}X = \text{temp}X + 1
\hspace{1cm} \text{if } \text{temp}X \leq Y \text{ then goto } k

\text{If } X = Y \text{ then goto } k
\hspace{1cm} 1 \text{ If } X < Y \text{ then goto } 4
\hspace{1cm} 2 \text{ If } Y < X \text{ then goto } 4
\hspace{1cm} 3 \text{ Go to } k
\hspace{1cm} 4 \text{ continue}

\text{For } i = 1 \text{ to } n
\hspace{1cm} 1 \text{ If } n = 0 \text{ then goto } 7
\hspace{1cm} 2 \text{ Let } i = 1
\hspace{1cm} \ldots
\hspace{1cm} 3 \text{ S}_1
\hspace{1cm} \ldots
\hspace{1cm} 4 \text{ S}_k
\hspace{1cm} \text{Next } i
\hspace{1cm} 5 \text{ Let } i = i + 1
\hspace{1cm} 6 \text{ If } n < i \text{ then goto } 3
\hspace{1cm} 7 \text{ continue}

\textbf{Theorem 1} \hspace{1cm} \text{The functions } Z = X + Y, \ Z = XY \ \text{and} \ Z = X^Y \ \text{are } UR-Basic \ \text{computable. The functions } X \dot{-} Y \ \text{is } UR-Basic \ \text{computable. The pair of functions remainder and quotient are } UR-Basic \ \text{computable i.e., input } n, m \ \text{then output } q, r \ \text{with } n = qm + r \ \text{and} \ 0 \leq r < m.\n
\textbf{Proof}\nZ = X + Y:\n\hspace{1cm} \text{Let } Z = X
\hspace{1cm} \text{For } i = 1 \text{ to } Y
\hspace{1cm} \hspace{1cm} \text{Let } Z = Z + 1
\hspace{1cm} \text{Next } i
\hspace{1cm} \text{Z = XY:}
Let \( Z = 0 \)
For \( i = 1 \) to \( Y \)
    Let \( Z = Z + X \)
Next \( i \)

\( Z = X^Y: \)
Let \( Z = 1 \)
For \( i = 1 \) to \( Y \)
    Let \( Z = ZX \)
Next \( i \)

\( Z = X^\frac{1}{Y}: \)
0 Let \( \text{saveY} = Y \)
1 If \( X \leq Y \) then goto 7
2 Let \( Z = X \)
3 If \( Y = 0 \) then goto 8
4 Let \( Y = Y - 1 \)
5 Let \( Z = Z - 1 \)
6 Go to 3
7 Let \( Z = 0 \)
8 Continue
9 Let \( Y = \text{saveY} \)

\( n = qm + r: \)
1 Let \( q = 0 \)
2 Let \( r = n \)
3 If \( r < m \) then goto 7
4 Let \( r = r - q \)
5 Let \( q = q + 1 \)
6 go to 3
7 continue

QED

**Hmwk 1.** (Fri 9-3) Prove that the greatest common divisor function \( d = \gcd(n, m) \) is UR-Basic computable. Or if you prefer the function \( f(n) \) = the \( n^{th} \) prime. Or you can prove that your favorite function is UR-Basic computable.
Primitive recursive functions

The class of primitive recursive functions is the smallest set of functions \( f : \omega^m \rightarrow \omega \) of arbitrary arity \( m \) which contain

1. the constant zero function, \( Z : \omega \rightarrow \omega \), \( Z(n) = 0 \) all \( n \),
2. the successor function, \( S : \omega \rightarrow \omega \) with \( S(n) = n + 1 \) all \( n \) (which we usually write \( n + 1 \)), and
3. the projections \( \pi^m_m(x_1, \ldots, x_n) = x_m \) for \( 1 \leq m \leq n < \omega \)

and is closed under

- composition: \( h \) is primitive recursive, if
  \[
h(x_1, \ldots, x_m) = f(g_1(x_1, \ldots, x_m), \ldots, g_n(x_1, \ldots, x_m))
  \]
  where \( f \) is \( n \)-ary and each \( g_i \) is \( m \)-ary are primitive recursive, and

- primitive recursion: \( h \) is primitive recursive, if
  \[
  \begin{align*}
  h(0, x_1, \ldots, x_m) &= g(x_1, \ldots, x_m) \\
  h(y + 1, x_1, \ldots, x_m) &= f(y, x_1, \ldots, x_m, h(y, x_1, \ldots, x_m))
  \end{align*}
  \]
  where \( g \) is \( m \)-ary and \( f \) is \((m + 2)\)-ary primitive recursive.

Note that by using the projections and compositions we may swap variables around and introduce dummy variables, e.g.

\[
h(x, y, z) = f(g(x, y), z, k(z, x)) = f(g_1(x, y, z), g_2(x, y, z), g_3(x, y, z))
\]

where

\[
\begin{align*}
g_1(x, y, z) &= g(\pi^3_1(x, y, z), \pi^3_2(x, y, z)) \\
g_2(x, y, z) &= \pi^3_3(x, y, z) \\
g_3(x, y, z) &= k(\pi^3_3(x, y, z), \pi^3_2(x, y, z))
\end{align*}
\]

A predicate \( P \subseteq \omega^n \) is primitive recursive iff its characteristic function \( \chi_P(\bar{x}) \) is where

\[
\chi_P(\bar{x}) = \begin{cases} 
1 & \text{if } P(\bar{x}) \\
0 & \text{if } \neg P(\bar{x})
\end{cases}
\]
Constant functions of any arity are primitive recursive. E.g.

\[ f(x, y, z) = S(S(Z(\pi^3, 1(x, y, z)))) = 2. \]

Define \( z = x + y \):
- \( x + 0 = x \)
- \( x + (y + 1) = (x + y) + 1 \)

Define \( z = xy \):
- \( x0 = 0 \)
- \( x(y + 1) = xy + x \)

Define \( z = x^y \):
- \( x^0 = 1 \)
- \( x^{y+1} = x^y x \)

Define \( z = x^{(y)} = x^{x^{x^{\ldots^{x}}}} \):
- \( x^{(0)} = x \)
- \( x^{(y+1)} = x^{x^{(y)}} \)

Define \( z = x! \):
- \( 0! = 1 \)
- \( (x + 1)! = (x + 1)x! \)

Define \( z = x^{-1} \):
- \( 0^{-1} = 0 \)
- \( (x + 1)^{-1} = x \)

Define \( z = y^{-x} \):
- \( y^{-0} = y \)
- \( y^{-}(x + 1) = (y^{-}x)^{-}1 \)

Define

\[ \text{sign}(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \end{cases} \]

by \( \text{sign}(x) = 1^{-}(1^{-}x) \).
Proposition 2  The predicates $x \leq y$, $x = y$, $x < y$ are primitive recursive.
If $P$ and $Q$ are primitive recursive predicates, then so is $P \lor Q$ and $\neg P$.
If $P(\vec{x}, y)$ is a primitive recursive predicate and $f(\vec{x})$ a primitive recursive
function, then $Q(\vec{x}) \equiv P(\vec{x}, f(\vec{x}))$ is a primitive recursive predicate.

Proof
\begin{align*}
\chi_{\leq}(x, y) &= 1 - (x - y) \\
\chi_{P \lor Q} &= \text{sign}(\chi_P + \chi_Q) \\
\chi_{\neg P} &= 1 - \chi_P \\
x = y &\iff x \leq y \text{ and } y \leq x \\
x < y &\iff \neg y \leq x \\
\chi_Q(\vec{x}) &= \chi_P(\vec{x}, f(\vec{x}))
\end{align*}
QED

Proposition 3  If $P(\vec{x}, y)$ is a primitive recursive predicate and $f(\vec{x})$ a primitive recursive
function, then
\begin{align*}
\exists y \leq f(\vec{x}) \ P(\vec{x}, y) \text{ and } \forall y \leq f(\vec{x}) \ P(\vec{x}, y)
\end{align*}
are both primitive recursive predicates.

Proof
Let
\begin{align*}
Q(\vec{x}, z) \equiv \exists y \leq z \ P(\vec{x}, y)
\end{align*}
Then $\chi_Q$ has the recursive definition:
\begin{align*}
\chi_Q(\vec{x}, 0) &= \chi_P(\vec{x}, 0) \\
\chi_Q(\vec{x}, z + 1) &= \text{sign}(\chi_Q(\vec{x}, z) + \chi_P(\vec{x}, z + 1))
\end{align*}
Note that
\begin{align*}
Q(\vec{x}, h(\vec{x})) &\equiv \exists y \leq h(\vec{x}) \ P(\vec{x}, y)
\end{align*}
and
\begin{align*}
\forall y \leq h(\vec{x}) \ P(\vec{x}, y) &\equiv \neg \exists y \leq h(\vec{x}) \ P(\vec{x}, y)
\end{align*}
QED

For example,
\begin{align*}
x \text{ divides } y &\iff \exists z \leq y \ y = xz. \\
x \text{ is a Prime} &\iff x > 1 \text{ and } \forall y \leq x \text{ if } y \text{ divides } x, \text{ then } y = 1 \text{ or } y = x.
\end{align*}
are primitive recursive predicates.
Bounded search: define $f(\vec{x}, z) = \mu y \leq z P(\vec{x}, y)$ where $f$ is the least $y \leq z$ which satisfies $P(\vec{x}, y)$ and $f = 0$ if no $y \leq z$ can be found.

**Proposition 4** Suppose $Q$ is a primitive recursive predicate and $h$ a primitive recursive function. Then

$$g(\vec{x}) = \mu y \leq h(\vec{x}) \ P(\vec{x}, y)$$

is primitive recursive.

**Proof**
Let

$$Q(\vec{x}, y) \equiv P(\vec{x}, y) \land \forall u < y \neg P(\vec{x}, u).$$

Then if we define

$$f(\vec{x}, z) = \mu y \leq z P(\vec{x}, y)$$

then

$$f(\vec{x}, z) = \sum_{y=0}^{z} y \cdot \chi_Q(\vec{x}, y)$$

which has the following primitive recursive definition:

- $f(\vec{x}, 0) = \chi_Q(\vec{x}, 0)$
- $f(\vec{x}, z + 1) = f(\vec{x}, z) + \chi_Q(\vec{x}, z + 1)$

Hence

$$g(\vec{x}) = f(\vec{x}, h(\vec{x})) = \mu y \leq h(\vec{x}) \ P(\vec{x}, y).$$

QED

**Proposition 5** If $f : \omega \to \omega$ is primitive recursive, the $\text{graph}(f)$ is a primitive recursive predicate. If $\text{graph}(f)$ is a primitive recursive predicate and there is a primitive recursive function $g$ which bounds $f$, then $f$ is primitive recursive.

**Proof**
Graph($f$) has characteristic function $\chi_=(\vec{x}, f(\vec{x}))$. If $f$ is bounded by $g$ then

$$f(\vec{x}) = \mu y \leq g(\vec{x}) \ P(\vec{x}, y)$$

is in the graph of $f$.

QED

Examples:
\[ z = \max(x, y) \text{ iff } (x = z \text{ and } x \geq y) \text{ or } (y = z \text{ and } y \geq x) \]

has primitive recursive graph and is bounded by \( x + y \), so it is a primitive recursive function.

Division, Quotient: input \( n, m > 0 \) output \( q, r \) with \( n = qm + r \) and \( r < m \). 
\( q = \text{quotient}(n, m) \) and \( r = \text{remainder}(n, m) \) both have primitive recursive graphs bounded by \( n + m \) so they are primitive recursive.

**Hmwk 2.** (Wed 9-8) Let \( r(n) = n^{th} \) digit of \( \sqrt{2} = 1.4142136 \ldots \), so \( r(0) = 1, r(1) = 4, \) and so on. Prove that \( r \) is primitive recursive. If you prefer you may use \( e = 2.7182818 \ldots \) instead of \( \sqrt{2} \). Does every naturally occurring constant in analysis have this property?

Coding pairs and sequences.

Coding pairs. \( \langle x, y \rangle = 2^x(2y + 1) - 1 \) is a bijection between \( \omega^2 \) and \( \omega \).
Both unpairing functions are primitive recursive since if \( x = \langle x_0, x_1 \rangle \), then \( x_0, x_1 \leq x \).
Triples can be coded by \( \langle x, y, z \rangle = \langle x, \langle y, z \rangle \rangle \) and similarly \( n \geq 4 \)-tuples.
To code arbitrary length finite sequences we can use primes:
Define: \( \text{nextprime}(x) = \mu y \leq x! + 1 \ y > x \text{ and } y \text{ is prime} \)
Note that if there is no prime between \( x \) and \( x! \) then \( x! + 1 \) is prime. Actually there is always a prime between \( x \) and \( 2x \).
Define: \( p_0 = 2 \) and \( p_n = \text{the } n^{th} \text{ odd prime, primitively recursively by:} \)
\( p_0 = 2 \)
\( p_{n+1} = \text{nextprime}(p_n) \).
Sequences are coded by \( c : \omega \times \omega \rightarrow \omega \) where
\( c(y, i) = \mu k \leq y \ p_i^{k+1} \) does not divide \( y \)
We often use \( y_i \) to denote \( c(y, i) \).

**Theorem 6** Every primitive recursive function is UR-Basic computable.

Proof
The empty program with input \( x \) and output \( y \), computes the constant zero function. Similarly for the projections. The successor function is computed by the one-line program “Let \( x = x + 1 \)”, with input and output variable \( x \).

For closure under composition: \( z = f(g_1(\bar{x}), \ldots, g_n(\bar{x})) \) use the basic program:
Let \( z_1 = g_1(\bar{x}) \)
Let \( z_2 = g_2(\bar{x}) \)
Let $z_n = g_n(\vec{x})$
Let $y = f(z_1, \ldots, z_n)$

where appropriate substitution of UR-Basic code has been done.

The basic code for a primitive recursion looks like
Let $z = g(\vec{x})$
For $i = 1$ to $y$
Let $z = h(i, z, \vec{x})$
next $i$

QED

Theorem 7 (Kleene) There exists a primitive recursive predicate $Q(e, x, y)$ and primitive recursive $g$ such that for every partial UR-Basic computable $f : \omega \rightarrow \omega$ there exists an $e$ with

$$f(x) = g(\mu y \ Q(e, x, y)).$$

Proof
We can assume that the UR-Basic program only uses the variable $v_i$ for $i < \omega$ and that the input variable is $v_0$ and output variable $v_1$.

1. $S = \langle 0, i \rangle \in \omega$ codes the statement “Let $v_i = v_i + 1$”.
2. $S = \langle 1, i \rangle \in \omega$ codes the statement “Let $v_i = v_i - 1$”.
3. $S = \langle n, i, j, k \rangle$ for $n \geq 2$ codes the statement “If $v_i \leq v_j$ then goto $k$”.

For $e \in \omega$ let $e = \langle n, S \rangle$ and let $S_0, S_1, \ldots, S_{n-1}$ be the program statements with $S_i$ coded by $c(S, i)$.

Next we define three primitive recursive predicates:
In the tuple $(e, x, y)$, $e$ codes the program, $x$ is the input value and $y$ is pair $\langle k, V \rangle$ coding the line $k$ in the program which is being executed and $V$ coding the values of the variables.

$$Init(e, x, y) \equiv$$
$$\exists V < y \ y = \langle 0, V \rangle \text{ and } c(V, 0) = x \text{ and } \forall i < e \ (i > 0 \rightarrow c(V, i) = 0)$$

Since this is the start we want to start with Statement 0, i.e., $y = (0, V)$ and $v_0 = x$ and $v_i = 0$ for all $i$ with $0 < i < e$. Note that we can bound this by $e$ since $e$ cannot refer to any variables with index higher than $e$.

$$Halt(e, y) \equiv \exists n, S < e \ \exists k, V < y \ y = \langle k, V \rangle \text{ and } e = \langle n, S \rangle \text{ and } k \geq n$$
All this says is we halt when we try to execute a line of the program beyond its length.

Onestep\((e, y, y')\)

This just says we take one step in executing the program. So it will be:

\[ \exists k, V, k', V' < y + y' \text{ and } \exists n, S < e \text{ such that all of the following are true:} \]

1. \( y = \langle k, V \rangle, \ y' = \langle k', V' \rangle, \text{ and } \ e = \langle n, S \rangle \)
2. \( k < n \) (we don’t take a step if program has halted)
3. If \( c(S, k) \) codes “Let \( v_i = v_i + 1 \)” then
   \[ c(V, i) = c(V', i) + 1, \]
   \[ c(V, j) = c(V', j) \text{ for all } j < e \text{ with } j \neq i, \text{ and} \]
   \[ k' = k + 1, \]
4. If \( c(S, k) \) codes “Let \( v_i = v_i - 1 \)” then
   \[ c(V, i) = c(V', i) - 1, \]
   \[ c(V, j) = c(V', j) \text{ for all } j < e \text{ with } j \neq i, \text{ and} \]
   \[ k' = k + 1. \]
5. If \( c(S, k) \) codes “If \( v_i \leq v_j \) then goto \( l \)” then
   \[ V = V' \text{ and} \]
   \[ k' = l \text{ if } c(V, i) \leq c(V, j) \text{ or } k' = k + 1 \text{ if } c(V, i) > c(V, j). \]

\( Q(e, x, y) \)

This says that \( y \) codes a computation using program \( e \) and input \( x \).

\[ Q(e, x, y) \equiv \]
\[ \exists N, Y < y \ y = \langle N, Y \rangle \text{ and } Init(e, x, c(Y, 0)) \text{ and } \text{Halt}(e, c(Y, N - 1)) \text{ and} \]
\[ \forall i < N \text{ Onestep}(e, c(Y, i), c(Y, i + 1)) \]

The function \( g \) simply extracts the value of \( v_1 \) the output variable from the computation sequence \( y \). Since \( g(y) \leq y \) it is enough to see that its graph is primitive recursive:
\[ g(y) = v \text{ iff } \exists N, Y, V, k < y \langle N, Y \rangle = y \text{ and } c(Y, N - 1) = \langle k, V \rangle \text{ and } c(V, 1) = v \]

QED

**Hmwk 3.** (Fri 9-11) Prove that there exists a (total) \( f : \omega \to \omega \) whose graph is a primitive recursive predicate but \( f \) is not a primitive function.

**Hmwk 4.** (Mon 9-13) Prove there exists a primitive recursive bijection \( f : \omega \to \omega \) such that \( f^{-1} \) is not primitive recursive.

**Corollary 8** The family of (partial) UR-Basic computable functions is the same as the family of (partial) recursive functions.

Church-Turing Thesis:
Every intuitively computable function is recursive.

**Proposition 9** There exists a computable function \( f : \omega \to \omega \) which is not primitive recursive.

**Proof**
Make an effective list \( f_n : \omega^k \to \omega \) of all the primitive recursive functions.
Define \( f(n) = f_n(n) + 1 \) if \( f_n \) is a 1-ary function, otherwise put \( f(n) = 0 \).
Since the listing can be effectively done by the Church-Turing Thesis the function \( f \) is recursive.
QED

**Proposition 10** There exists a universal partial computable function
\[ \psi : \omega \to \omega \]

i.e. if we define \( \psi_e(x) = \psi(\langle e, x \rangle) \) then \( \{ \psi_e : e \in \omega \} \) is a uniformly computable listing of all partial recursive functions.
Proof
\[ \psi((e, x)) = g(\mu y \, Q(e, x, y)). \]
QED

**Proposition 11** *(Padding Lemma)* There exists a 1-1 computable function \( p \) such that \( \psi_e = \psi_{p(e, n)} \) for every \( e, n \).

*(S-n-m Theorem).* There exists a computable function \( S : \omega^2 \to \omega \) such that \( \psi_{e_0}(e_1, x) = \psi_{S(e_0, e_1)}(x) \) for all \( e_0, e_1, x \).

Proof
To pad the program \( S_0, S_1, \ldots, S_m \) coded by \( e \) just add the statement

\[ S_{m+1} = \text{LetDonothing}(e, n) = \text{Donothing}(e, n) + 1 \]

and let \( p(e, n) \) code this new program.

Given \( P \) the program coded by \( e_0 \) and input \( e_1 \) make-up a new program coded by \( S(e_0, e_1) \) which puts \( e_1 \) into \( P \)'s first input variable and then pops into program \( P \).

QED

This proposition can be used as follows: Suppose we have described a partial computable function \( \theta(e, x) \). Then there exists a one-to-one computable function \( f : \omega \to \omega \) such that

\[ \forall e, x \ \psi_{f(e)}(x) = \theta(e, x) \]

When use it this way we should call it the 1-1-S-1-1 Theorem.

**Definition 12** \( A \subseteq \omega \) is computably enumerable iff either \( A \) is empty or \( A \) is the range of a computable function, i.e., \( A = \{a_0, a_1, a_2, \ldots\} \) where the function \( n \mapsto a_n \) is computable. This is abbreviated c.e.

\( A \subseteq \omega \) is computable iff its characteristic function \( \chi_A \) is computable.

\( A \subseteq \omega \) is \( \Sigma_0^1 \) iff there exists a computable predicate \( R \subseteq \omega^2 \) such that \( A = \{n : \exists m \, R(n, m)\} \).

**Proposition 13** For \( A \subseteq \omega \) the following are equivalent:

(1) \( A \) is computably enumerable.

(2) \( A \) is the domain of a partial computable function.

(3) \( A \) is \( \Sigma_0^1 \).

(4) \( A \) is finite or \( A \) has a one-to-one computable enumeration.
Proof

(1) → (2): Given a computable enumerable listing $a_n$, describe a partial computable function $f$ by input $x$ and look for $x$ on the list. Halt if you find it, otherwise continue looking forever.

(2) → (1): Define $\psi_{e,s}(x) \downarrow = y$ to mean that $e, x, y < x$ and the $e^{th}$ UR-Basic program with input $x$ converges and outputs $y$ in fewer than $s$ steps. The predicate

$$P(e, x, y, s) \equiv \psi_{e,s}(x) \downarrow = y$$

is primitive recursive. If $A$ is the domain of $\psi_e$, then either $A$ is empty or let $x_0 \in A$ be arbitrary and define a recursive enumeration of $A$ by $a_n = x$ if $n = \langle x, y, s \rangle$ and $\psi_{e,s}(x) \downarrow = y$ otherwise $a_n = x_0$.

(1) → (3): Let $f : \omega \to \omega$ be computable and have range $A$. Let $R$ be the graph of $f$, then $y \in A$ iff $\exists x R(x, y)$.

(3) → (2): Suppose $x \in A$ iff $\exists y R(x, y)$. Then $f(x) = \mu y R(x, y)$ is partial recursive with domain $A$.

(1) → (4): Given $\{a_n : n < \omega\}$ a computable enumeration of $A$, define a computable enumeration $\{b_n : n < \omega\}$ by $b_{n+1} = a_m$ where $m$ is the least such that $a_m \notin \{b_i : i \leq n\}$.

QED

Definition 14 For $A \subseteq \omega$, $\overline{A} = \omega \setminus A$ the complement of $A$.

$A \subseteq \omega$ is $\Pi^0_1$ iff $\overline{A}$ is $\Sigma^0_1$.

$\Delta^0_1 = \Sigma^0_1 \cap \Pi^0_1$.

Proposition 15 For $A \subseteq \omega$ the following are equivalent:

(1) $A$ is computable.

(2) $A$ and $\overline{A}$ are both c.e.

(3) $A$ is $\Delta^0_1$.

(4) $A$ is finite or $A$ has a strictly increasing computable enumeration.

Proof

(1) → (2): Since computable implies c.e.

(2) → (1): Input $x$. Effectively list $A$ and $\overline{A}$ simultaneously until $x$ shows up.

(2) iff (3): Trivial

(1) → (4): Take $a_n$ to be the $n^{th}$ element of $A$.

(4) → (1): Let $\{a_n : n < \omega\}$ be a strictly increasing computable enumeration of $A$.
Input $x$. Find $n$ such that $a_n > x$. Then $x \in A$ iff $x \in \{a_i : i < n\}$.

QED

**Hmwk 5.** (Wed 9-15) Prove that every nonempty computably enumerable set $A$ is the range of a primitive recursive function. Extra Credit: prove that not every infinite computably enumerable set is the range of a one-to-one primitive recursive function.

**Proposition 16** Every infinite c.e. set contains an infinite computable set.

Proof
Given $\{a_n : n < \omega\}$ a computable enumeration of $A$, define a strictly increasing computable enumeration $\{b_n : n < \omega\}$ by $b_{n+1} = a_m$ where $m$ is the least such that $a_m > b_n$.

QED

**Proposition 17** If $A$ and $B$ are c.e. sets, then $A \cap B$ is c.e. and $A \cup B$ is c.e. If $A$ and $B$ are computable sets, then $A \cap B$, $A \cup B$, and $\overline{A}$ are all computable sets.

Proof
Domain of $f + g$ is the intersection of domain $f$ and domain $g$. Enumerate $A \cup B$ by $x_{2n} = a_n$ and $x_{2n+1} = b_n$.

QED

**Hmwk 6.** (Fri 9-17) Suppose that $V \subseteq \omega$ is c.e. For each $n$ define $V_n = \{x : \langle n, x \rangle \in V\}$. Prove that $\bigcup_n V_n$ is c.e.

**Example 18** There exists an c.e. set $K$ which is not computable.

Proof
$$K = \{e : \psi_e(e) \downarrow\}$$

If $\overline{K}$ is the domain of $\psi_e$, then $e \in K$ iff $e \notin K$.

QED

**Example 19** There exists disjoint c.e. sets $K_0$ and $K_1$ which are computably inseparable, i.e., there is not exists a computable set $R \subseteq \omega$ with $K_0 \subseteq R$ and $K_1 \subseteq \overline{R}$. 

14
\[ K_0 = \{ e : \psi_e(e) \downarrow = 0 \} \text{ and } K_1 = \{ e : \psi_e(e) \downarrow = 1 \} \]  

QED

**Definition 20** For any \( \Gamma \subseteq P(\omega) \) define \( \tilde{\Gamma} \) to be the set of all \( \tilde{A} \) for \( A \in \Gamma \) and define \( \Delta = \Gamma \cap \tilde{\Gamma} \). \( \text{Sep}(\Gamma) \) is the property that for every \( A, B \in \Gamma \) disjoint there exists \( C \in \Delta \) with \( A \subseteq C \) and \( B \subseteq \overline{C} \). \( \text{Red}(\Gamma) \) (the reduction principle) is the property that for every \( A, B \in \Gamma \) there exists disjoint \( A' \subseteq A \) and \( B' \subseteq B \) with \( A', B' \in \Gamma \) and \( A \cup B = A' \cup B' \).

**Proposition 21** \( \text{Red}(\Gamma) \) implies \( \text{Sep}(\tilde{\Gamma}) \).

Proof  
Apply reduction to the complements.  
QED

**Proposition 22** \( \text{Red}(\Sigma^0_1) \) and hence \( \text{Sep}(\Pi^0_1) \).

Proof  
\( A = \{ x : \exists u R(u, x) \} \) and \( B = \{ x : \exists v S(v, x) \} \). Put  
\[ x \in A' \leftrightarrow \exists u R(u, x) \text{ and } \forall v \leq u \neg S(v, x) \]  
\[ x \in B' \leftrightarrow \exists v S(v, x) \text{ and } \forall u < v \neg R(u, x) \]  
QED

In example 19 it follows that \( K_0 \) and \( K_1 \) cannot be separated by disjoint \( \Pi^0_1 \) sets \( B_0 \) and \( B_1 \) because such a \( B_0 \) and \( B_1 \) could be computably separated.

**Hmwk 7.** (Mon 9-20) Prove \( \text{Sep}(\Gamma) \) for \( \Gamma = \{ A \cup B : A \in \Sigma^0_1, B \in \Pi^0_1 \} \).

**Definition 23** \( A \leq_m B \) iff there exists a computable function \( f \) such that  
\[ \forall x \in \omega \ x \in A \leftrightarrow f(x) \in B. \]  
If the \( f \) can be taken one-to-one, then we write \( A \leq_1 B \).

Note that \( A \leq_m B \) and \( B \) is computable, then \( A \) is computable.

**Definition 24** \( W = \{ \langle e, x \rangle : \psi(\langle e, x \rangle) \downarrow \} \). Then \( \{ W_e : e \in \omega \} \) where \( W_e = \{ x : \langle e, x \rangle \in W \} \) is a uniform listing of the c.e. sets.
Example 25  Empty = \{e : W_e = \emptyset\} is not computable.

Proof
Define
\[
\theta(e, x) = \begin{cases} 
1 = 0 & \text{if } e \in K \\
\uparrow & \text{otherwise}
\end{cases}
\]

By the S-n-m theorem there exists \( f \) computable such that

\[
\forall e, x \quad \psi_{f(e)}(x) = \theta(e, x)
\]

But then \( e \in K \) iff \( W_{f(e)} \neq \emptyset \) iff \( f(e) \notin E \) so \( K \leq_m E \) and therefore \( E \) not computable.

QED

Proposition 26  (Rice) If \( A \) is a nontrivial index set, then \( A \) is not computable.

Proof
Like proof for Empty.

QED

Theorem 27  (Myhill) \( A \leq_1 B \) and \( B \leq_1 A \) iff there exists a computable bijection \( \pi : \omega \to \omega \) with \( \pi(A) = B \).

Proof
The Schroeder-Bernstein Theorem says: if there exists a 1-1 \( f : A \to B \) and 1-1 \( g : B \to A \), then there exists a bijection \( h : A \to B \). One way to prove this is to assume \( A \) and \( B \) are disjoint and define a bipartite graph on the vertices \( A \cup B \). Put \( a \in A \) connected to \( b \) iff either \( f(a) = b \) or \( g(b) = a \). As \( f \) and \( g \) are 1-1 the order of every vertex is either 1 or 2. The connected components of this graph come in 4 types, see figure 1. Note that in Type 1 the point \( a \in A \) is not in the range of \( g \) and in Type 2 the point \( b \in B \) is not in the range of \( f \). Type 4 components are infinite in both “directions” while Type 3 is the only finite component.

To get \( h \) simply define \( h = f \) on any component of type 1,3, or 4 and \( h = g^{-1} \) on components of type 2.

The proof of Myhill’s theorem is similar except we may never know exactly which type of component we looking at.

Suppose \( f \) and \( g \) are 1-1 computable functions reducing \( A \) to \( B \) and \( B \) to \( A \).

Effectively construct a sequence \( \pi_s \) of bijections with
Figure 1: Schroeder-Bernstein connected components
1. $\pi_s : D_s \to E_s$ is a bijection.
2. $D_s$ and $E_s$ are finite subsets of $\omega$.
3. $\pi_s \subseteq \pi_{s+1}$.
4. $n \in D_{2n}$ and $n \in E_{2n+1}$.
5. if $\pi_s(n) = m$, then either $m = f g f g \cdots f n$ or $n = g f g f \cdots g m$.

In the condition 5 we have dropped the parentheses to make it more readable.

If we then take $\pi = \bigcup_s \pi_s$ then $\pi$ is a recursive bijection since we effectively constructed the sequence. It takes $A$ to $B$, because suppose $\pi(n) = m$. Then if $m = f g f g \cdots f n$

$$n \in A \text{ iff } f n \in B \text{ iff } g f n \in A \text{ iff } f g f n \in B \text{ iff } \cdots \text{ iff } m = f g f g \cdots f n \in B$$

similarly if $m = g f g f \cdots g m$

$$m \in B \text{ iff } g m \in A \text{ iff } f g m \in B \text{ iff } g f g m \in A \text{ iff } \cdots \text{ iff } n = g f g f \cdots g m \in A$$

either way $n \in A$ iff $m \in B$.

At stage $s=0$ we take $\pi_0$ to be the empty function.

At stage $s+1$ suppose we are given $\pi_s : D_s \to E_s$. If $s = 2n$ we try to extend $\pi_s$ to include $n \in D_{s+1}$. If its already there we let $\pi_{s+1} = \pi_s$. Otherwise consider the following sequences:
Let \( n = n_0 \), \( f_0 = m_0 \) and in general \( f(n_k) = m_k \) and \( g(m_k) = n_{k+1} \), see figure 2.

Case 1. For some \( k \) we have that \( m_k \notin E_s \).

In this case we put \( \pi_{s+1} = \pi_s \cup \{<n_0, m_k>\} \).

Case 2. Not case 1.

In this case the connected component of the graph (see Figure 1) must be of Type 3, i.e., a finite closed loop. Suppose \( g(m_k) = n_0 \). But by condition 5 if all the \( m_k \) are in \( E_s \), then they must map via \( \pi_s^{-1} \) to the set \( \{n_0, n_1, \ldots, n_k\} \) (although not in any particular order). But this is a contradiction, since \( n = n_0 \notin D_s \). Hence Case 2 cannot happen.

The construction at stage \( s+1 \) where \( s = 2n + 1 \) is entirely analogous except we make sure \( n \in E_{s+1} \).

QED

**Theorem 28** (Rogers) Suppose \( \rho : \omega \to \omega \) is partial computable and we define \( \rho_e(x) = \rho(e, x) \). Suppose

1. \( \rho \) is universal, i.e., \( \{\rho_e : e \in \omega\} \) includes all partial computable functions.

2. \( \rho \) satisfies padding, i.e., there exists one-to-one computable \( p : \omega \times \omega \to \omega \) such that

\[
\forall e, n \rho_e = \rho_{p(e,n)}
\]

3. \( \rho \) satisfies S-1-1, i.e., there exists a computable \( S : \omega \times \omega \to \omega \) such that

\[
\forall e_1, e_2, x \rho_{e_1}((e_2, x)) = \rho_{S(e_1, e_2)}(x)
\]

Then there exists a computable bijection \( \pi : \omega \to \omega \) such that

\[
\forall e \psi_e = \rho_{\pi(e)}
\]

Proof

Let \( \psi = \rho_{e_0} \). Using padding and S-1-1 for \( \rho \) we can find a 1-1 computable function \( f(e) = p(S(e_0, e)) \) such that

\[
\forall e \psi_e = \rho_{S(e_0, e)} = \rho_{f(e)}
\]
similarly there is a 1-1 computable function $g$ such that

$$\forall e \rho_e = \psi_{g(e)}.$$

By the proof of Theorem 27 there is a computable bijection $\pi : \omega \to \omega$ with the property that whenever $\pi(n) = m$ then either $m = f g f g \cdots f n$ or $n = g f g f \cdots g m$. But

$$\psi_n = \rho_{fn} = \psi_{gfn} = \cdots = \rho_{fgfg \cdots fn} = \rho_m$$

and

$$\rho_m = \psi_{gm} = \rho_{fgm} = \cdots = \psi_{fgfg \cdots gm} = \psi_n$$

so in either case $\psi_n = \rho_{\pi(n)}$.

QED

**Hmwk 8.** (Wed 9-22) Find an example of a partial computable $\rho$ which is universal but fails to satisfy padding. Find an example which is universal, satisfies padding but fails to satisfy S-1-1. (S-1-1 implies padding see Soare p.25-26.)

**Theorem 29 (Kleene - Recursion Theorem)** For any computable function $f$ there exists an $e$ with $\psi_e = \psi_{f(e)}$.

**Proof**

Define a partial computable function $\theta$ by

$$\theta(u, x) = \psi_{\psi_u(u)}(x) = \psi((\psi(u, u), x))$$

By padding-S-1-1 we can find a (one-to-one) computable function $d : \omega \to \omega$ such that

$$\forall u \ \psi_{d(u)}(x) = \theta(u, x)$$

Let $v$ be an index for $f \circ d$, i.e.,

$$\forall x \ \psi_v(x) = f(d(x))$$

Put $e = d(v)$ then

$$\psi_e(x) = \psi_{d(v)}(x) = \theta(v, x) = \psi_{\psi_v(v)}(x) = \psi_{f(d(v))}(x) = \psi_{f(e)}(x)$$

QED
From the proof we can get an infinite computable set of fixed points \( e \),

since we can take any \( v' \) such that \( \psi_{v'} = f \circ d \) and set \( e' = d(v') \). Also note that our fixed point \( e \) is obtained effectively from an index for \( f \), so given a computable \( f : \omega \times \omega \to \omega \) if we let \( f_n : \omega \to \omega \) be defined by \( f_n(x) = f(n, x) \) then we get a fixed points \( e_n \)

\[
\psi_{e_n} = \psi_{f_n(e_n)}
\]

and the function \( n \mapsto e_n \) is computable. This is called the recursion theorem with parameters:

\[
\forall n \quad \psi_{e(n)} = \psi_{f(n,e(n))}.
\]

**Example 30** There are infinitely many \( e \) such that \( \psi_e(0) = e \). There are infinitely many \( e \) such that \( W_e = \{e\} \).

Proof

Define \( \theta(e, x) = e \) for all \( e \). By the S-n-m Theorem there exists a computable \( f \) such that

\[
\forall e, x \quad \psi_f(e) = \theta(e, x)
\]

By the Recursion Theorem there are infinitely many fixed points for \( f \), i.e.,

\[
\psi_e = \psi_f(e)
\]

and for each of these \( \psi_e \) is the constant function \( e \).

Define a partial computable function \( \theta \) by

\[
\theta(e, x) = \begin{cases} 
1 = 0 & \text{if } e = x \\
\uparrow & \text{otherwise}
\end{cases}
\]

By S-n-m theorem there is a computable function \( g \) with \( \psi_g(e)(x) = \theta(x) \).

By the definition of \( \theta \) we see that for every \( e \):

\[
W_{g(e)} = \{e\}
\]

By the Recursion Theorem there are infinitely many fixed points for \( g \) and for any of them

\[
W_e = W_{g(e)} = \{e\}.
\]

**Hmwk 9.** (Fri 9-24) Prove:

(a) for every \( f, g \) computable functions, there exists \( e_1 \) and \( e_2 \) such that

\[
\psi_{f(e_1)} = \psi_{e_2} \quad \text{and} \quad \psi_{g(e_2)} = \psi_{e_1}
\]

(b) \( \exists e_1 \neq e_2 \quad W_{e_1} = \{e_2\}, \ W_{e_2} = \{e_1\} \)

(c) \( \exists e_1 > e_2 > e_3 \quad W_{e_1} = \{e_2\}, \ W_{e_2} = \{e_3\}, \ W_{e_3} = \{e_1\} \)
Example 31 (Smullyan) For any computable functions \( f(x,y) \) and \( g(x,y) \) there exists \( a, b \in \omega \) such that

\[
\psi_{f(a,b)} = \psi_a \quad \text{and} \quad \psi_{g(a,b)} = \psi_b
\]

Proof

By the recursion theorem

\[
\forall x \exists y \ \psi_{g(x,y)} = \psi_y
\]

but since the fixed point \( y \) is obtained effectively from \( x \) and an index for \( g \) there exists a computable function \( h \) such that

\[
\forall x \ \psi_{g(x,h(x))} = \psi_{h(x)}
\]

Apply the fixed point theorem to \( f(x,h(x)) \) there exists \( a \in \omega \) such that

\[
\psi_{f(a,h(a))} = \psi_a
\]

Letting \( b = h(a) \) does the job.

QED

Hmwk 10. (Mon 9-27) Prove

(a) \( \exists e_1 < e_2 < e_3 \) \( W_{e_1} = \{e_2\}, W_{e_2} = \{e_3\}, W_{e_3} = \{e_1\} \)

(b) \( \exists e_1 \neq e_2 \) \( W_{e_1} = \{e_1,e_2\} = W_{e_2} \)

(c) \( \exists e_1 < e_2 < e_3 \) \( W_{e_1} = \{e_2,e_3\}, W_{e_2} = \{e_1,e_3\}, W_{e_3} = \{e_1,e_2\} \)

Definition 32 A c.e. set \( A \) is m-complete iff \( B \leq_m A \) for every c.e. \( B \). Similarly 1-complete. Define \( C \) is creative iff \( C \) is c.e. and there exists a computable function \( q \in \omega^\omega \) such that for every \( e \)

\[
W_e \cap C = \emptyset \rightarrow q(e) \notin C \cup W_e.
\]

Theorem 33 (Myhill) For \( C \subseteq \omega \) c.e. the following are equivalent:

1. \( C \) is creative
2. \( C \equiv_1 K \)
3. \( C \) is 1-complete
4. \( C \) is m-complete
Proof

(2) → (3): It is enough to see that $K$ is 1-complete, since then for any $B$ c.e. we would have $B \leq_1 K \leq_1 A$. Define a partial computable function $\rho$ as follows:

$$\rho(e, x) = \begin{cases} \downarrow = 0 & \text{if } e \in B \\ \uparrow & \text{otherwise} \end{cases}$$

$\rho$ is partial computable because we enumerate $B$ looking to see if $e$ ever turns up, if not the computation never halts. Using the 1-1-S-1-1 Theorem there exists a 1-1 computable function $f$ such that

$$\forall e, x \ \psi_{f(e)}(x) = \rho(e, x) = \begin{cases} \downarrow = 0 & \text{if } e \in B \\ \uparrow & \text{otherwise} \end{cases}$$

Then $e \in B$ iff $\psi_{f(e)}(f(e)) \downarrow$ iff $f(e) \in K$.

(3) → (4): Trivial

(4) → (1): The creativity of $K$ is witnessed by the identity function, i.e.,

$$W_e \cap K = \emptyset \rightarrow e \notin W_e \cup K.$$ 

Suppose $K \leq_m A$ is witnessed by the function $f$. Then there exists a computable function $g$ such that

for all $e$ $W_{q(e)} = f^{-1}(W_e)$

(Use S-1-1 to get $\psi_{q(e)} = \psi_e \circ f$.) Then

$$W_e \cap A = \emptyset \rightarrow$$

$$f^{-1}(W_e) \cap K = \emptyset \rightarrow$$

$$W_{q(e)} \cap K = \emptyset \rightarrow$$

$$q(e) \notin f^{-1}(W_e) \cup K \rightarrow$$

$$f(q(e)) \notin W_e \cup A$$

so $f \circ q$ witnesses the creativity of $A$.

(1) → (2):

**Claim** The creativity function for $A$ can be taken to be 1-1.

**Proof**

Given any creativity function $d$ for $A$. Construct a computable function $f$ such that

$$\forall x \ W_{f(x)} = W_x \cup \{d(x)\}.$$
To do this use
\[ \forall x, y \psi_{f(x)}(y) = \rho(x, y) = \begin{cases} \downarrow = 0 & \text{if } y \in W_x \text{ or } y = d(x) \\ \uparrow & \text{otherwise} \end{cases} \]

Now we get our 1-1 creativity function \( \hat{d} \) recursively as follows: Input \( e \) put \( e = e_0 \) and effectively generate the sequence \( e_{s+1} \) where \( W_{e_{s+1}} = W_{e_s} \cup \{ d(e_s) \} \), i.e. put \( e_{s+1} = f(e_s) \). and look for \( e_s \) such that. Simultaneously enumerate \( A \) and \( W_e \) looking for something in their intersection.

Search for the least \( s \) such that either
1. \( e_s > \hat{d}(e - 1) \) or
2. \( A_s \cap W_{e_s} \neq \emptyset \)

If the first happens put \( \hat{d}(e) = e_s \). If the second happens put \( \hat{d}(e) = \hat{d}(e - 1) + 1 \).

This proves the Claim.

QED

Now we show that \( K \leq_1 A \). Define a partial computable function \( \theta \) as follows:
\[ \psi_{f(n,x)}(y) = \theta(n, x, y) = \begin{cases} \downarrow = 0 & \text{if } n \in K \text{ and } y = \hat{d}(x) \\ \uparrow & \text{otherwise} \end{cases} \]

It follows that
\[ W_{f(n,x)} = \begin{cases} \{ \hat{d}(x) \} & \text{if } n \in K \\ \emptyset & \text{otherwise} \end{cases} \]

By the uniform proof of the recursion theorem and by padding we get a 1-1 computable sequence \( n \mapsto e_n \) of fixed points so that
\[ \forall n \ W_{f(n,e_n)} = W_{e_n} = \begin{cases} \{ \hat{d}(e_n) \} & \text{if } n \in K \\ \emptyset & \text{otherwise} \end{cases} \]

But then \( n \in K \) iff \( \hat{d}(e_n) \in A \). So \( K \leq_1 A \).

QED

Most naturally occurring noncomputable c.e. sets are m-complete.

**Hmwk 11.** (Wed 9-29) Prove or disprove: there exists a computable function \( d : \omega \to \omega \) such that for every \( e \)
\[ W_e \cap K \text{ finite } \rightarrow d(e) \notin W_e \cup K \]
**Definition 34** \(A\) is simple iff \(A\) is c.e., \(\overline{A}\) is infinite, and \(\overline{A}\) does not contain an infinite c.e. set.

**Theorem 35** (Post) There exists a simple set.

Proof
Define a computable sequence \(A_s \subseteq s\) of increasing finite sets as follows. \(A_0 = \emptyset\). At stage \(s + 1\) find the least \(e < s\) (if any) such that \(W_{e,s} \cap A_s = \emptyset\) and \(\exists x > 2e \in W_{e,s}\). Put \(A_{s+1} = A_s \cup \{x\}\) for the least \(e\) and \(x\) for which this is true. If this happens we say that \(e\) has acted at stage \(s + 1\). If there no such \(e\), then put \(A_{s+1} = A_s\).

The set \(A = \cup_s A_s\) is simple. Note that each \(e\) can act at most once. Hence if \(W_e\) is infinite and \(W_e \cap A = \emptyset\), eventually there will come a stage \(s\) where \(\exists x > 2e \in W_{e,s}\) and all smaller \(e\)'s which will ever act have already acted at a previous stage. But then \(e\) will act, which is a contradiction.

Also we see that \(\overline{A}\) is infinite because for all \(e\) \(|A \cap 2e| \leq e\) since the only \(e^{pr}\) which can put an \(x\) into \(A\) with \(x \leq 2e\) are those \(e'\) with \(e' < e\).

QED

**Definition 36** \(A \leq_T B\) or \(A\) is Turing reducible to \(B\). Add to the UR-Basic programming language statements of the form:

\[
\text{Let } y = \chi_B(x)
\]

for any variables \(x, y\). This programming language is called Oracle UR-Basic. Then \(A \leq_T B\) iff there is an Oracle UR-Basic program with Oracle for \(B\) which computes the characteristic function \(\chi_A\) of \(A\).

**Hmwk 12.** (Fri 10-1) Suppose \(A\) is a simple set and \(A = \{a_n : n \in \omega\}\) is a 1-1 computable enumeration of \(A\). Prove there exists infinitely many \(n\) such that \(W_{a_n} = \{a_m : m > n\}\). (Hint: it is easier to show there exists \(e \in A\) such that \(W_e = \{e\}\).)

**Proposition 37** (Dekker Deficiency Set) For every c.e. set \(A\) which is not computable there exists a simple set \(B\) with \(B \equiv_T A\).

Proof
Let \( \{a_n : n \in \omega\} \) be a 1-1 computable enumeration of \( A \). Define

\[
B = \{ n : \exists m > n \ a_m < a_n \}
\]

It is easy to see that \( B \) is c.e.

\( B \) is infinite: Otherwise there would be an \( N \) such that \( a_{n+1} > a_n \) for all \( n > N \) and then \( A \) would be computable.

\( A \leq_T B \): Input \( x \). Find \( n \in B \) such that \( a_n > x \). Then \( x \in A \) iff \( x \in \{a_i : i < n\} \).

\( B \) does not contain an infinite computable set: Suppose \( R \subseteq B \) is an infinite computable set. But then the argument we just gave for \( A \leq_T B \) shows that \( A \leq_T R \) which would make \( A \) computable.

\( B \leq_T A \): Input \( n \). Using an Oracle for \( A \) check if

\[
\{a_i : a_i < a_n \text{ and } i < n\} = A \cap \{x : x < a_n\}
\]

if they are equal, then \( n \notin B \), otherwise \( n \in B \).

QED

**Hmwk 13.** (Mon 10-4) Define \( B \subseteq \omega \) is intro-reducible iff \( B \leq_T C \) for every infinite \( C \subseteq B \). Prove that for every \( A \) there exists \( B \equiv_T A \) intro-reducible.

**Definition 38** For \( A \subseteq \omega \) define the Turing degree of \( A \) to be

\[
a = \text{deg}(A) = \{ B : B \equiv_T A \}.
\]

Let \( D = \{\text{deg}(A) : A \subseteq \omega\} \) be the Turing Degrees. \( (D, \leq) \) is the partial order where \( a \leq b \) iff \( A \leq_T B \).

**Definition 39** For \( \sigma \in 2^{<\omega} \text{ and } e, x, y, s \in \omega \) we write

\[
\{e\}^\sigma_s(x) \downarrow = y
\]

to mean that the \( e \)th oracle machine with input \( x \) and using \( \sigma \) to answer Oracle questions, converges in less than \( s \) steps and outputs \( y \). We also require that \( e, x, y < s \) and that in this computation the oracle is not asked about \( n \notin \text{dom}(\sigma) \) or \( n \geq s \).

**Proposition 40** The predicate \( O(\sigma, e, x, y, s) \) defined by

\[
O(\sigma, e, x, y, s) \text{ iff } \{e\}^\sigma_s(x) \downarrow = y
\]

is primitive recursive.
Definition 41  For $A \subseteq \omega$ the jump of $A$ is defined by

$$A' = \{ e : \exists s \ e_s^A(x) \downarrow \}$$

Proposition 42  (1) $A \leq_T B$ implies $A' \leq_1 B'$.

(2) $A <_T A'$

Proof
(1) Define

$$\theta(e, x) = \begin{cases} \downarrow = 0 & \text{if } e^A(e) \downarrow \\ \uparrow & \text{otherwise} \end{cases}$$

Then $\theta$ is partial computable in $A$ and since $A \leq_T B$ we have that $\theta$ is partial computable in $B$. By the 1-1-S-1-1 Theorem relativized to $B$ there exists a 1-1 computable function $f$ such that

$$\forall e, x \ \{ f(e) \}^B(x) = \theta(e, x).$$

But then $e \in A'$ iff $\{ e \}^A(e) \downarrow$ iff $\{ f(e) \}^B(f(e)) \downarrow$ iff $f(e) \in B'$.

(2) To see $A \leq_1 A'$ construct a 1-1 computable function $f$ so that $f(n)^A(?)$ has the same computation on any input and it converges iff $n \in A$. Then $n \in A$ iff $f(n) \in A'$. To see that $A' \not\leq_T A$, suppose that it is. Define $f = 1 - \chi_{A'}$. Then since $f \leq_T A' \leq_T A$ there is an $e_0$ with $\{ e_0 \}^A = f$. But then $e_0 \in A'$ iff $e_0 \notin A'$.

QED

Corollary 43  If $A \equiv_T B$, then $A' \equiv_T B'$. Hence, letting $a' \in \mathcal{D}$ be the Turing degree of $A'$ is well-defined and $a < a'$ for every $a \in \mathcal{D}$.

Similarly, $a''$ is the jump of the jump of $a$, and $a^{(n)}$ is $n$ jumps of $a$.

Definition 44  $a \mid b$ iff not $a \leq b$ and not $b \leq a$. I.e. the degrees $a$ and $b$ are Turing incomparable.

Proposition 45  (Kleene-Post) There exists $a, b \in \mathcal{D}$ with $a \mid b$.

Proof
Construct sequences $(\sigma_s \in 2^{<\omega} : s \in \omega), (\tau_s \in 2^{<\omega} : s \in \omega)$ with the property that $\sigma_s \subseteq \sigma_{s+1}$ and $\tau_s \subseteq \tau_{s+1}$ for each $s$. For $s = 0$ take $\tau_s$ and $\sigma_s$ to be the empty sequence.
At stage $s + 1$ we are given $\tau_s$ and $\sigma_s$ and we do as follows:

Case $s = 2e$:
Let $n = |\tau_s|$.  

Case a. There exists $\sigma \supseteq \sigma_s$ such that $\{e\}^\sigma(n) \downarrow$. In this case put $\sigma_{s+1} = \sigma$ and put $\tau_{s+1} = \tau_si$ where $i = 0, 1$ whichever is different from $\{e\}^\sigma(n)$.

Case b. No such $\sigma$. Put $\sigma_{s+1} = \sigma_s$ and put $\tau_{s+1} = \tau_s0$.

Case $s = 2e + 1$:
Let $n = |\sigma_s|$ and proceed similarly to $s = 2e$ with the roles of $\sigma_s$ and $\tau_s$ reversed.

This ends the construction. We put $A = \cup_{s \in \omega} \sigma_s$ and $B = \cup_{s \in \omega} \tau_s$.

QED

It is easy to see that the entire construction is computable in $\theta'$ and hence there are incomparable Turing degrees beneath $\theta'$.

**Proposition 46 (Kleene-Post)** For every $a \in D \setminus \{o\}$ there exists $b \in D$ with $a \mid b$.  

Let $deg(A) = a$. Construct $(\tau_s \in 2^{<\omega} : s \in \omega)$ as follows. $\tau_0 = \langle \rangle$.

At stage $s + 1$ we are given $\tau_s$.

Case $s = 2e$. Let $n = |\tau_s|$. Take $i = 0$ or $i = 1$ so that $i \neq \{e\}^A(n)$. Put $\tau_{s+1} = \tau_si$.

Case $s = 2e + 1$.

Case a. There exists $n < \omega$, $\rho_1, \rho_2$ with $\tau_s \subseteq \rho_i$ and

\[
\{e\}^{\rho_1}(n) \downarrow \neq \{e\}^{\rho_2}(n) \downarrow
\]

In this case we put $\tau_{s+1} = \rho_1$ or $\tau_{s+1} = \rho_2$ which ever that case is that

\[
\{e\}^{\tau_{s+1}}(n) \neq A(n).
\]

Case b. There is no such $n$ and $\rho_i$. Put $\tau_{s+1} = \tau_s0$.

This ends the construction. Now we check that $B = \cup_s \tau_s$ is Turing incomparable to $A$. The cases $2e$ easily show that it is not the case that $B \leq_G TA$. Suppose $A \leq_T B$ and choose $e$ so that $\{e\}^B = A$ and consider
stage $s+1$ where $s = 2e+1$. In case (a) we get that $\{e\}^B(n) \neq A(n)$ so that it is impossible. Now we show that case (b) cannot happen. Define

$$f(n) = i \text{ iff } \exists \tau \supseteq \tau_s \{e\}^T(n) \downarrow = i$$

Note that $f$ is well-defined because we are in case (b) and $f$ is total because we assume that $\{e\}^B$ is the characteristic function of $A$. Hence $f$ which is computable is the characteristic function of $A$, which contradicts the assumption that $A$ is not computable.

QED

Hmwk 14. (Wed 10-6) Prove that for every countable $A \subseteq D \setminus \{0\}$ there exists $b \in D$ such that $a \upharpoonright b$ for all $a \in A$.

Definition 47 $A \oplus B = \{2n : n \in A\} \cup \{2n+1 : n \in B\}$.

Proposition 48 $A_0 \leq_T A_1$ and $B_0 \leq_T B_1$ implies $A_0 \oplus B_0 \leq_T A_1 \oplus B_1$.

$A \leq_T C$ and $B \leq_T C$ iff $A \oplus B \leq_T C$.

Definition 49 $a \lor b = \deg(A \oplus B)$ is the join or least upper bound of $a$ and $b$.

Meets, $a \land b$, in the Turing degrees may or may not exist.

Proposition 50 (Kleene-Post) There exists $a, b \in D \setminus \{0\}$ with $a \land b = 0$ i.e., for all $c$ if $c \leq a$ and $c \leq b$ then $c = 0$.

Proof

As before construct sequences $(\sigma_s \in 2^{<\omega} : s \in \omega)$, $(\tau_s \in 2^{<\omega} : s \in \omega)$ with the property that $\sigma_s \subseteq \sigma_{s+1}$ and $\tau_s \subseteq \tau_{s+1}$ for each $s$. For $s = 0$ take $\tau_s$ and $\sigma_s$ to be the empty sequence.

At stage $s+1$ we are given $\tau_s$ and $\sigma_s$ and we do as follows:

Case $s = 3e$. Let $n = |\sigma_s|$. Let $i = 0$ or $i = 1$ so that $\psi_e(n) \neq i$. Put $\sigma_{s+1} = \sigma_s i$.

Case $s = 3e + 1$. Similar to $3e$ but for $\tau_{s+1}$.

Case $s = 3\langle e_1, e_2 \rangle + 2$. 


Case a. There exists $n < \omega$, $\sigma \supseteq \sigma_s$, and $\tau \supseteq \tau_s$ such that

\[ \{e_1\}^\sigma(n) \downarrow \neq \{e_2\}^\tau(n) \downarrow \]

put $\sigma_{s+1} = \sigma$ and $\tau_{s+1} = \tau$.

Case b. Not case a. Put $\tau_{s+1} = \tau_s$ and $\sigma_{s+1} = \sigma_s$.

This ends the construction. We put $A = \cup_s \sigma_s$ and $B = \cup_s \tau_s$. The stages $3e, 3e + 1$ guarantee that neither $A$ nor $B$ is computable. Now suppose that $C \leq_T A$ and $C \leq_T B$. This will be witnessed by a pair $e_1$ and $e_2$. At stage $s = 3\langle e_1, e_2 \rangle + 2$ it must have been that Case a. failed since we assume that

\[ \{e_1\}^A = \{e_2\}^B = C. \]

But then we may define a total computable function $f$ by

\[ f(n) = i \text{ iff } \exists \sigma \supseteq \sigma_s \ {e_1}^\sigma(n) \downarrow = i \]

and $f$ must be the characteristic function of $C$ and hence $C$ is computable.

QED

**Proposition 51** (Kleene-Post) For every $c \in \mathcal{D}$ there exists $a, b \in \mathcal{D}$ with $a \wedge b = c$ and $a \mid b$, i.e., $a > c$, $b > c$, and for all $d$ if $d \leq a$ and $d \leq b$ then $d \leq c$.

Proof

This is a relativization of the above argument. Construct $A_0$ and $B_0$ so that for every $e$

\[ \{e\}^C \neq A_0 \oplus C \text{ and } \{e\}^C \neq B_0 \oplus C \]

and

\[ \{e_1\}^{A_0 \oplus C} = \{e_2\}^{B_0 \oplus C} = D \rightarrow D \leq_T C \]

Then take $A = A_0 \oplus C$ and $B = B_0 \oplus C$.

QED

**Hmwk 15.** (Fri 10-8) Prove that for every $c \in \mathcal{D}$ there exists $a, b \in \mathcal{D}$ with $a \mid b$, $a \wedge b = 0$, and $a \lor b \geq c$. Hint: one way to code $C$ into $A \oplus B$ is to use boot-strapping. Define

\[ x_{2n} = \mu x > x_{2n-1} \ A(x) = 1 \]

\[ x_{2n+1} = \mu x > x_{2n} \ B(x) = 1 \]

\[ n \in C \text{ iff } x_n \text{ is even} \]
Proposition 52  (Spector) Given \( (a_n : n < \omega) \) in \( D \) with \( a_n < a_{n+1} \) for all \( n \) there exists \( b, c \in D \) with

1. \( a_n \leq b \) and \( a_n \leq c \) for all \( n \) and
2. for all \( d \in D \) if \( d \leq b \) and \( d \leq c \) then there exists \( n \) with \( d \leq a_n \).

Proof
Let \( \deg(A_n) = a_n \) and set \( A = \{ \langle n, x \rangle : n < \omega, x \in A_n \} \). The key to this construction is to make \( B \) and \( C \) have the property that for each \( n \) \( B_n =^* A_n =^* C_n \) where \( B_n = \{ x : \langle n, x \rangle \in B \} \) and \( C_n = \{ x : \langle n, x \rangle \in C \} \).

As before construct sequences \( (\sigma_s \in 2^{<\omega} : s \in \omega), (\tau_s \in 2^{<\omega} : s \in \omega) \) with the property that \( \sigma_s \subseteq \sigma_{s+1} \) and \( \tau_s \subseteq \tau_{s+1} \) for each \( s \). For \( s = 0 \) take \( \tau_s \) and \( \sigma_s \) to be the empty sequence.

At stage \( s + 1 \) we will extend \( \sigma_s \) and \( \tau_s \) so as to agree with \( A_i \) for \( i < s \) on new elements of their domain. Define

\[
\begin{align*}
    f_s &= \sigma_s \cup \{ \langle \langle i, x \rangle, j \rangle : \langle i, x \rangle \notin \text{dom}(\sigma_s), \ i < s, \ \text{and} \ A_i(x) = j \} \\
    g_s &= \tau_s \cup \{ \langle \langle i, x \rangle, j \rangle : \langle i, x \rangle \notin \text{dom}(\tau_s), \ i < s, \ \text{and} \ A_i(x) = j \}
\end{align*}
\]

Note that \( f_s \) is a partial function extending \( \sigma_s \) which agrees with the characteristic function of each \( A_i \) for \( i < s \) except possible on the (finite) domain of \( \sigma_s \). Similarly \( g_s \).

Let \( s = \langle e_1, e_2 \rangle \).

Case a. There exists \( n < \omega, \sigma \supseteq \sigma_s \) and \( \tau \supseteq \tau_s \) such that \( f_s \cup \sigma \) is a function (i.e., they are compatible) and \( g_s \cup \tau \) is a function and

\[
\{e_1\}^\sigma(n) \downarrow \neq \{e_2\}^\tau(n) \downarrow
\]

Put \( \sigma_{s+1} = \sigma \) and \( \tau_{s+1} = \tau \).

Case b. Not Case a. Put \( \sigma_{s+1} = \sigma_s \) and \( \tau_{s+1} = \tau_s \).

This completes the construction, so put \( B = \cup_s \sigma_s \) and \( C = \cup_s \tau_s \).

Claim. For all \( n \) we have that \( A_n \leq_T B \) and \( A_n \leq_T C \). To see this note that in the construction that for all \( s > n \) that \( f_s(\langle n, m \rangle) = f_{s+1}(\langle n, m \rangle) \).

Furthermore, except for the finitely many element of the domain of \( \sigma_{n+1} \) we
have that \( A_n(m) = f_{n+1}(n,m) \). It follows that \( A_n =^* B_n \) and so \( A_n \leq_T B_n \leq_T B \). Similarly for \( C \).

Claim. Suppose that \( D \leq_T B \) and \( D \leq_T C \). Then \( D \leq_T A_n \) for some \( n < \omega \). To see this suppose that

\[
\{e_1\}^B = \{e_2\}^C = D
\]

and \( s = (e_1, e_2) \). Since the characteristic functions of \( B \) and \( C \) extend \( \sigma_{s+1} \) and \( \tau_{s+1} \) respectively it is evident that Case (a) could not have occurred. So we assume Case (b). Note that in this case it is impossible that there exists \( n, \rho_1, \rho_2 \) with \( \sigma_s \subseteq \rho_1 \) and \( \sigma_s \subseteq \rho_2 \), and each of \( \rho_1 \) and \( \rho_2 \) compatible with \( f_s \) such that

\[
\{e_1\}^{\rho_1}(n) \downarrow \neq \{e_1\}^{\rho_2}(n) \downarrow .
\]

This is because \( \{e_2\}^C(n) \downarrow \) and so then we would be in Case (a).

It follows easily as before that \( D = \{e_1\}^B \leq_T f_s \). But

\[
f_s \leq_T A_0 \oplus A_1 \oplus \cdots \oplus A_{s-1} \leq_T A_{s-1}
\]

so \( D \leq_T A_{s-1} \).

QED

**Proposition 53 (Friedberg Jump Inversion)** For every \( a \in D \) if \( a \geq d' \) then there exists \( b \in D \) with \( b' = a \).

**Proof**

We construct sequence \( (\tau_s : s \in \omega) \) computable in \( A \oplus 0' \equiv_T A \) as follows.

At stage \( s + 1 \) we are given \( \tau_s \in 2^{\omega} \)

(a) We put \( \tau = \tau_s i \) where \( i = A(s) \).

(b) Let \( e = s \). We ask \( 0' \) if there exists \( \sigma \supseteq \tau \) such that

\[
\{e\}_{|\sigma|}(e) \downarrow
\]

If there is such a \( \sigma \) then we effectively find one and put \( \tau_{s+1} = \sigma \).

More precisely, before the construction begins find a computable function \( f(e, \tau) \) such that

1. for any \( e, \tau \)

\[
\psi_{f(e,\tau)}(0) \downarrow \iff \exists \sigma \supseteq \tau \ \{e\}_{|\sigma|}(e) \downarrow
\]

32
2. when \( \psi_{f(e, \tau)}(0) \) converges it outputs such a \( \sigma \) and

3. the algorithm \( \psi_{f(e, \tau)}(\)?) ignores its input.

We put \( \tau_{s+1} = \tau \) if \( f(e, \tau) \notin 0' \), otherwise we put \( \tau_{s+1} = \sigma = \text{def} \, \psi_{f(e, \tau)}(0) \).

This ends the construction. We let \( B = \cup_{s \in \omega} \tau_s \).

Claim.

1. \( (\tau_s : s \in \omega) \leq_T A \oplus 0' \leq_T A \)
2. \( A \leq_T (\tau_s : s \in \omega) \)
3. \( (\tau_s : s \in \omega) \leq_T B \oplus 0' \)
4. \( B' \leq_T (\tau_s : s \in \omega) \)

Proof

1. The construction only requires oracles for \( 0' \) and \( A \). Also \( A \geq_T 0' \).
2. We encoded the characteristic function of \( A \) at step (a). Hence

\[
s \in A \text{ iff } \tau_{s+1}(|\tau_s|) = 1.
\]

3. Recursively construct the sequence \( (\tau_s : s \in \omega) \) using oracles for \( 0' \) and \( B \). Given \( \tau_s \) we use that \( \tau_{s+1} \subseteq B \) to figure out the first digit, i.e., \( \tau \) of step (a). To do step (b) we only used \( 0' \) and the computable function \( f \).
4. By our construction given any \( e \) let \( s = e \), then we have that

\[
e \in B' \text{ iff } \{e\}^B(e) \downarrow \text{ iff } \{e\}[^{\tau_{s+1}}_{|\tau_{s+1}|}] \downarrow
\]

This proves the Claim. But note that the Claim implies

\[
B' \leq_T (\tau_s : s \in \omega) \leq_T A \leq_T (\tau_s : s \in \omega) \leq_T B \oplus 0' \leq_T B'
\]

QED

Hmwk 16. (Mon 10-11) Prove that \( \forall a \in \mathcal{D} \ a \geq o' \rightarrow \exists b, c \in \mathcal{D} \ b|c \text{ and } b' = a = c' \).

Theorem 54 (Clifford Spector) There exists a minimal Turing degree, i.e., \( \exists a \in \mathcal{D} \text{ with } a < a \text{ but no } b \in \mathcal{D} \text{ with } a < b < a. \)
Proof
For any $\sigma \in 2^n$, i.e., a finite sequence of zeros and ones, we can code $\sigma$ by the number
\[ x = 2^n + \sum \{2^i : i < n \text{ and } \sigma(i) = 1 \}. \]
The extra $2^n$ is there to distinguish sequences ending in zeros from each other. We suppress this coding and just talk about computable subsets of $2^{<\omega}$.

**Definition 55** $T \subseteq 2^{<\omega}$ is a perfect tree iff
\begin{enumerate}
  \item $T$ is nonempty,
  \item $\sigma \subseteq \tau \in T$ implies $\sigma \in T$, and
  \item $\forall \sigma \in T \exists \tau_0, \tau_1 \in T$ with $\sigma \subseteq \tau_0, \sigma \subseteq \tau_1$, and $\tau_0$ and $\tau_1$ are incomparable.
\end{enumerate}

**Definition 56** For $T \subseteq 2^{<\omega}$ a tree we define:
\begin{enumerate}
  \item $\sigma \in T$ splits iff $\sigma 0, \sigma 1 \in T$
  \item $\sigma = \text{stem}(T)$ iff $\sigma$ splits but no shorter node of $T$ splits
  \item $[T] = \{x \in 2^\omega : \forall n \ x|n \in T\}$
  \item for $\sigma \in T$ let $T(\sigma) = \{\tau \in T : \tau \subseteq \sigma \text{ or } \sigma \subseteq \tau\}$
\end{enumerate}

To prove the Theorem construct a sequence $(T_s : s \in \omega)$ of computable perfect trees as follows.

At stage $s = 0$ take $T_0 = 2^{<\omega}$.

At stage $s + 1$ where $s = 2e$ let $\sigma = \text{stem}(T_s)$ and $n = |\sigma|$. If $\psi_e(n) \downarrow = 0$ then put $T_{s+1} = T_s(\sigma 1)$ otherwise put $T_{s+1} = T_s(\sigma 0)$.

At stage $s + 1$ where $s = 2e + 1$ we obtain $T_{s+1} \subseteq T_s$ a perfect computable subtree as follows. We first ask the question:

Does there exist $\sigma \in T_s$ such that for all $\sigma_1, \sigma_2 \in T(\sigma)$ and $n, m_1, m_2 < \omega$ if $\{e\}^{\sigma_1}(n) \downarrow = m_1$ and $\{e\}^{\sigma_2}(n) \downarrow = m_2$, then $m_1 = m_2$?
Case (a) If the answer is yes, we take $T_{s+1} = T_s(\sigma)$ for any such $\sigma$.

Case (b) If the answer is no, we construct computable sequences 

$$(\sigma_\rho \in T : \rho \in 2^{<\omega})$$

and

$$(n_\rho \in \omega : \rho \in 2^{<\omega})$$

such that

1. $\{e\}^{\sigma_{\rho_0}}(n_\rho) \downarrow \neq \{e\}^{\sigma_{\rho_1}}(n_\rho) \downarrow$ and
2. $\sigma_\rho \subseteq \sigma_{\rho_0}$ and $\sigma_\rho \subseteq \sigma_{\rho_1}$.

Note that (1) implies that $\sigma_{\rho_0}$ is incomparable to $\sigma_{\rho_1}$. We put

$$T_{s+1} = \{ \sigma : \exists \rho \in 2^{<\omega} \sigma \subseteq \sigma_\rho \}$$

then $T_{s+1}$ is a computable perfect subtree of $T_s$.

This ends the construction of the sequence of trees. Note that $T_{s+1} \subseteq T_s$.

Take $A$ to be the subset of $\omega$ whose characteristic function is the unique element of $\cap_{s \in \omega}[T_s]$. It is easy to see that stage $2e + 1$ guarantees that $A$ is not computable, so it is enough to see stage $2e + 2$ guarantees that if $B = \{e\}^A$ then either $B$ is computable or $A \leq_T B$.

Case (a) for all $\sigma_1, \sigma_2 \in T_{s+1}$ and $n, m_1, m_2 < \omega$ if $\{e\}^{\sigma_1}(n) \downarrow = m_1$ and $\{e\}^{\sigma_2}(n) \downarrow = m_2$, then $m_1 = m_2$. In this case $B$ is computable, since $A \in [T_{s+1}]$ and $B = \{e\}^A$ means that all we have to do to compute $B(n)$ is to search the computable tree $T_{s+1}$ for any $\sigma$ for which $\{e\}^{\sigma}(n) \downarrow$ and then $B(n) = \{e\}^{\sigma}(n)$.

Case (b) In this case we show that $A \leq_T B$. We know $A \in [T_{s+1}]$. Suppose we know that $\sigma_\rho \subseteq A$. To decide whether $\sigma_{\rho_0} \subseteq A$ or $\sigma_{\rho_1} \subseteq A$, we compute both of

$$\{e\}^{\sigma_{\rho_0}}(n_\rho) \text{ and } \{e\}^{\sigma_{\rho_1}}(n_\rho).$$

Since these two computations are guaranteed to converge and to different values at most one of them can agree with $B(n_\rho)$. One of them must agree and so using an oracle for $B$ we can determine the unique $i = 0, 1$ so that $\sigma_{\rho_i} \subseteq A$.

QED

Hmwk 17. (Fri 10-15) Prove that there are uncountably many minimal degrees.
Theorem 57 (Sacks) Minimal upper bounds exists. Given any sequence of degrees \((a_n \in D : n < \omega)\) such that \(a_n < a_{n+1}\) for all \(n\) there exists \(b \in \mathcal{D}\) with \(a_n < b\) all \(n\) but there is no \(c \in \mathcal{D}\) with \(a_n < c < b\) for all \(n\).

Proof

Here we use the notion of a computably-pointed tree.

Definition 58 \(T \subseteq 2^{<\omega}\) is computably-pointed iff \(T\) is a perfect tree and \(T \leq_T A\) for every \(A \in [T]\).

The new ingredient required in this construction is

Claim. Suppose \(T \subseteq 2^{<\omega}\) is computably-pointed tree and \(T \leq_T B\). Then there exists \(T^* \subseteq T\) a computably-pointed tree such that \(T^* \equiv_T B\).

Proof

There exists a natural bijection \(f : 2^{<\omega} \rightarrow \text{Split}(T)\) where \(\text{Split}(T)\) are the splitting nodes of \(T\). Note that \(f\) and \(T\) are Turing equivalent. Given \(B \in 2^\omega\) let

\[ T_B = \{ \sigma \in 2^{<\omega} : \sigma(2n) = B(n) \text{ whenever } 2n < |\sigma| \}. \]

Now take \(T^*\) to be the tree generated by \(f(T_B)\).

QED

Construct \((T_s : s \in \omega)\) a sequence of computably-pointed trees as follows.

Suppose \(T_s \equiv_T A_s\) and \(e = s\). Relativizing Spector’s proof above to \(T_s\) we can obtain \(T^o \subseteq T_s\) with \(T^o \leq_T T_s\) a perfect subtree so that for every \(B \in [T^o]\): if \(C = \{e\}^B\) then either \(B \leq_T (C \oplus T^o)\) or \(C \leq_T T^o\).

Note that \(T^o\) is computably-pointed and \(T^o \leq_T A_s\). Hence by applying the Claim above we can obtain \(T_{s+1} \subseteq T^o\) such that \(T_{s+1}\) is computably-pointed and \(T_{s+1} \equiv_T A_{s+1}\).

This ends the construction. We let \(B\) be the unique element of \(\cap_{s \in \omega}[T_s]\).

First note that \(A_s \leq_T B\) for each \(s\), because \(B \in [T_s]\), \(T_s\) is computably-pointed and so \(A_s \equiv_T T_s \leq_T B\).

Suppose that \(A_s \leq_T C \leq_T B\) for every \(s \in \omega\). Then at some stage \(s = e\) we have that \(C = \{e\}^B\). Hence by construction either \(C \leq_T T^o \leq_T A_s\) or \(B \leq_T (C \oplus T^o)\). The first is impossible since \(A_s <_T A_{s+1} \leq_T C\) and so it must be that \(B \leq_T (C \oplus T^o)\). But \(T^o \leq_T A_s \leq_T C\) so \(B \leq_T C\).

QED
Hmwk 18. (Mon 10-18) (a) Prove there exists \( a, b \in D \) with \( o < a < b \) and not there exists \( c \) with either \( o < c < a \) or \( a < c < b \).

(b) (Extra Credit) Prove there exists \( a, b \in D \) with \( o < a < b \) and \( c \leq b \) iff \( c = 0 \) or \( c = a \) or \( c = b \), for all \( c \in D \).

**Definition 59** The use of an oracle computation \( \{e\}^A(x) \) written

\[
\text{use}(\{e\}^A(x))
\]

is \( n+1 \) where \( n \) is the maximum number for which the oracle for \( A \) is queried.

Note that if \( u = \text{use}(\{e\}^A(x)) \) and \( B \cap u = A \cap u \) then \( \{e\}^A(x) \) and \( \{e\}^B(x) \) are the same computation.

**Theorem 60** (Friedberg-Muchnik) There exists c.e. sets \( A_0 \) and \( A_1 \) such that \( A_0 \not\leq_T A_1 \) and \( A_1 \not\leq_T A_0 \).

**Proof**
Our requirements are:

\[
R_{2e+i} \quad \{e\}^{A_i} \neq A_{1-i}
\]

for each \( e \in \omega \) and \( i = 0, 1 \).

The strategy for meeting this requirement is to attach a follower \( x \in \omega \) to \( R_{2e+i} \) and then wait until \( \{e\}^{A_{i,s}}(x) \downarrow = 0 \). When this happens we put \( x \) into \( A_{1-i} \) and try to avoid injuring the computation \( \{e\}^{A_{i,s}}(x) \). If we succeed then \( \{e\}^{A_i}(x) = 0 \neq 1 = A_{1-i}(x) \). If we wait forever, then \( x \) is never put into \( A_{1-i} \) and so \( A_{1-i}(x) = 0 \neq \{e\}^{A_i}(x) \). In either case the requirement \( R_{2e+i} \) is met.

There are two possible successful outcomes for this strategy, either we wait forever or we act at some stage and then preserved the relevant computation.

**Construction**

Everything in the construction will be done effectively.

At each stage \( s \) of the construction we will have effectively constructed:

1. finite sets \( A_{i,s} \) for \( i = 0, 1 \),
2. a follower \( x = x_{q,s} \) for each \( R_q \) with \( q < s \), and
3. a function \( f_s \) with domain \( s \) which is attempting to predicate the final outcomes of our strategy for each \( R_q \) with \( q < s \).
At stage $s = 0$ put $A_{i,0} = \emptyset$ for $i = 0,1$. Nobody has followers and $f_s$ is the empty function.

At stage $s + 1$ look for the least $q = 2e + i < s$ such that

1. $f_s(q) =$‘waiting’ and
2. $\{e\}_s^{A_{i,s}}(x) \downarrow = 0$ with use less than $s$ where $x = x_{q,s}$ is the follower of $R_{2e+i}$.

If we find such a $q$ then we take the following actions:

1. Put $x$ into $A_{1-i}$, i.e.,
   \[ A_{1-i,s+1} = A_{1-i,s} \cup \{x\} \]
2. Set $f_{s+1}(q) =$‘acted’.
3. Reappoint followers for lower priority requirements, i.e. for each $q' > q$ with $q' < s + 1$ put $x = \langle q', s + 1 \rangle$ to be the follower of $R_{q'}$.
4. Make all lower priority requirements start over, i.e., for each $q' > q$ put $f_{s+1}(q') =$‘waiting’.

We say that $R_q$ acted at stage $s + 1$. If there is no such $q$ then we just continue to wait. In either case assign $x = (s, s + 1)$ to be the follower of $R_s$ and put $f_{s+1}(s) =$‘waiting’.

This ends the stage and the construction.

Note that the sequence
\[ (A_{s,0}, A_{s,1}, f_s, x_{q,s} : s \in \omega, q < s) \]

is computable.

We put $A_i = \cup_{s \in \omega} A_{i,s}$. These are c.e. sets since $A_{i,s} \subseteq A_{i,s+1}$.

Verification

Claim. For each $q$

1. $R_q$ acquires a permanent follower, i.e., there exist some stage $s_0$ such that for all $s > s_0$ the follower of $R_q$ at stage $s$ is that same as at stage $s_0$. 38
2. $R_q$ is met, i.e., $\{e\}^{A_i} \neq A_{1-i}$

3. $R_q$ acts at most finitely many times.

Proof
This is the main claim and it is proved by induction on $q$.

So suppose that (3) is true for all $q' < q$. Then there is a stage $s_0$ such that some $q' < q$ acted and no such $q' < q$ acts after stage $s_0$. Then the follower $x_q$ of $R_q$ appointed at stage $s_0$ is the permanent follower of $R_q$. Furthermore $f_{s_0}(q) = \text{‘waiting’}$.

Suppose $q = 2e + i$. After stage $s_0$ there are two possibilities:
(a) for some $s > s_0$ we have that $\{e\}_{s}^{A_i,s}(x_q) \downarrow = 0$ with use less than $s$ or
(b) not (a).

Suppose (a). In this case since no higher priority $q'$ acts after stage $s_0$ then $R_q$ will act. Hence $x_q$ is put into $A_{1-i}$. Furthermore all other followers of lower priority requirements appointed now or at future stages will be larger than the use of the computation $\{e\}_{s}^{A_i,s}(x_q)$ (we assume that $s \leq \langle q', s \rangle$).

Hence
$$\{e\}^{A_i}(x_q) \downarrow = 0 \neq 1 = A_{1-i}(x_q)$$

Suppose (b). In this case it must be that either
$$\{e\}^{A_i}(x_q) \uparrow \text{ or } \{e\}^{A_i}(x_q) \downarrow \neq 0.$$ 

In either case $x_q$ is never put into $A_{1-i}$ - this is because the possible followers of two distinct requirements are disjoint and no follower is used again for the same requirement. So $A_{1-i}(x_q) = 0 \neq \{e\}^{A_i}(x_q)$ and thus $R_q$ is met.

So as we see $R_q$ will act at most one more time after stage $s_0$ and so it acts only finitely many times. This proves the Claim and the Theorem.

QED

We say that $R_q$ is injured when it is made to appoint new followers and start over. Hence, the terminology ‘finite injury priority argument’.

Corollary 61 There exists a set $A$ which is c.e. and $0 <_T A <_T 0'$.

Proof
Since $0$ and $0'$ are $\leq_T$ comparable to every c.e. set it must be that both $A_i$ from the Friedberg-Muchnik Theorem are strictly in between.

QED
Another way to prove that some c.e. degree is nontrivial is to construct a
low simple set \( A \). Since a simple set is not computable we have that \( 0 <_T A \).
Low means that \( A' \equiv_T 0' \) so \( A <_T 0' \) by Lemma 42.

**Lemma 62 (The Limit Lemma)** Suppose \( g \in \omega^\omega \), then
\[
\text{if } g \leq_T 0' \text{ then there exists } f : \omega \times \omega \to \omega \text{ computable such that for all } n
\]

\[
\lim_{s \to \infty} f(n, s) = g(n)
\]

**Proof**
Suppose \( g = \{e\}^0 \). Let \( (0'_s : s \in \omega) \) be a computable enumeration of \( 0' \),
e.g., \( 0'_s = \{e < s : \{e\}_s(e) \downarrow\} \). Define

\[
f(n, s) = \begin{cases} 1 & \text{if } \{e\}_s^0(n) \downarrow \\ 0 & \text{otherwise} \end{cases}
\]

Then \( g(n) = \lim_{s \to \infty} f(n, s) \).

For the converse, suppose that \( g(n) = \lim_{s \to \infty} f(n, s) \) where \( f \) is computable. For each \( n \) using an oracle for \( 0' \) we can compute \( s_0 \) so that for every \( s > s_0 \) we have that \( f(n, s) = f(n, s_0) \).

(Try \( s_0 = 0 \) and ask the oracle if the computation that searches for a change in \( f \) ever terminates. If yes, try \( s_0 = 1 \), etc. Continue incrementing \( s_0 \) until the oracle says that beyond this stage \( f \) does not change.)

It follows that \( g(n) = f(n, s_0) \). Hence there is an algorithm with oracle \( 0' \) which computes \( g \).

QED

**Theorem 63** There exists a low simple set \( A \), i.e. \( A' \equiv 0' \) and \( A \) is simple.

**Proof**
We make \( A \) simple by a strategy that is suggested by the proof of the limit
lemma, namely we would like to use

\[
f(e, s) = \begin{cases} 1 & \text{if } \{e\}_s^{A'}(e) \downarrow \\ 0 & \text{otherwise} \end{cases}
\]

to show that \( A' \leq_T 0' \). That is, \( A'(e) = \lim_{s \to \infty} f(e, s) \). If \( e \in A' \) then it is
easy to see that \( f(e, s) = 1 \) for all sufficiently large \( s \). The problem then is
to make sure that if \( f(e, s) = 1 \) for infinitely many \( s \), then \( e \in A' \).
So we make the following requirements:

\[ Ne \quad (\exists s \quad \{e\}^A_s(e) \downarrow \rightarrow \{e\}^A(e) \downarrow) \]

In order to make sure that the set \( A \) is simple we have the following requirements:

\[ Pe \quad (W_e \text{ infinite}) \rightarrow W_e \cap A \neq \emptyset \]

The strategy for \( Pe \) is the same as for the Post Simple Set construction (Theorem 35), that is we wait for some \( x \in W_e,s \) with \( x > 2e \) and \( A_s \cap W_e,s = \emptyset \) and put \( x \) into \( A_{s+1} \).

The strategy for \( Ne \) is to wait until we see convergence and then try to prevent the computation from changing by restraining numbers less than the use of the computation from entering \( A \).

The requirement \( Pe \) is positive since the strategy is try to put things into \( A \) while the requirement \( Ne \) is negative since it tries to keep things out of \( A \).

**Construction**

At each stage in the construction we will have \( A_s \) and \( r(e, s) \) for each \( e \). We will always have that \( r(e, s) = 0 \) for \( e \geq s \) so the function \( r \) is really a finite function.

Stage \( s + 1 \). Look for the least \( e < s \) such that

1. \( W_{e,s} \cap A_s = \emptyset \)
2. \( \exists x > 2e \) with \( x \in W_{e,s} \) and \( x > r(e', s) \) for all \( e' < e \).

For the least such \( e \) choose the least \( x \) as above and put \( A_{s+1} = A_s \cup \{x\} \). We say in this case that \( Pe \) acted at stage \( s + 1 \). If there is no such \( e \) put \( A_{s+1} = A_s \).

Next we compute \( r(e, s + 1) \) for all \( e < s + 1 \). If \( \{e\}^A_{s+1}(e) \downarrow \), then put

\[ r(e, s + 1) = \text{use}(\{e\}^A_{s+1}(e)) \]

otherwise put \( r(e, s + 1) = 0 \).

This is the end of the construction. We let \( A = \cup_{s \in \omega} A_s \) which is c.e.

**Verification.**

**Claim.**

1. \( Pe \) is met.

41
2. \( N_e \) is met.

3. \( \lim_{s\to\infty} r(e, s) = r(e) < \infty \) exists.

Proof

We prove this by induction on \( e \). Note that each \( P_e \) can act at most once, since after it acts \( W_e \) and \( A \) are no longer disjoint. Assume the claim is true for every \( e' < e \).

(1) By induction we have some \( s_0 \) such that for all \( s > s_0 \) and \( e' < e \) that \( r(e', s) = r(e') \). Put

\[
R = \max\{r(e') : e' < e\}.
\]

We can also choose \( s_0 \) so large that no \( P_{e'} \) for \( e' < e \) acts after stage \( s_0 \) since each \( P_{e'} \) acts at most once. Suppose that \( W_e \) is infinite. It follows that at some stage \( s > s_0 \) there will be a \( x \in W_{e,s} \) such that \( x > 2e + R \). At stage \( s + 1 \) either \( A_s \cap W_{e,s} \neq \emptyset \) or \( P_e \) will act. In either case \( P_e \) is met.

(2) Choose \( s_0 \) so that no \( P_{e'} \) for \( e' \leq e \) acts after stage \( s_0 \). This means that after stage \( s_0 \) no positive requirement can ever injure a computation of \( N_e \). Hence if there is some \( s_1 > s_0 \) such that \( \{e\}_{s_1}^{A_{s_1}}(e) \downarrow \) then no \( x < \text{use}\{e\}_{s_1}^{A_{s_1}}(e) \) will ever enter \( A \). It follows that this is the final computation and therefore \( \{e\}_{s_1}^{A_{s_1}}(e) \downarrow \) with the same computation as at stage \( s \).

(3) As above, either we never see convergence and then \( r(e, s) = 0 \) for all \( s > s_0 \) or we see convergence and then \( r(e, s) = r(e, s_1) \) for all \( s > s_1 \).

This finishes the proof of the Claim and the Theorem.

QED

Hmwk 19. (Fri 10-22) (From Soare) A set \( A \) is auto-reducible iff there exists \( e \) such that for every \( x \) we have

\[
\{e\}_{A_e \setminus \{x\}}(x) \downarrow = A(x).
\]

Prove that there exists a \( A \) low c.e. set which is not auto-reducible.

We define

\[
A_n = \{x : \langle n, x \rangle \in A\}
\]

and

\[
\oplus_{k \neq n} A_k = \{ \langle k, x \rangle : k < \omega \text{ and } k \neq n \}.
\]
Theorem 64 There exists an c.e. set $A$ such that for every $n$

$$A_n \not\leq_T \oplus_{k \neq n} A_k$$

Proof
This is a minor modification of the Friedberg-Muchnic argument (Theorem 60).

Our requirements are:
$$R_{\langle e, n \rangle} \{e\}^{\oplus_{k \neq n} A_k} \neq A_n$$
for $e, n \in \omega$. And the construction is nearly the same:
At stage $s + 1$ look for the least $q = \langle e, n \rangle < s$ such that

1. $f_s(q) = \text{‘waiting’}$ and

2. $\{e\}^{\oplus_{k \neq n} A_k, s} (x) \downarrow = 0$ with use less than $s$ where $x = x_{q, s}$ is the follower of $R_q$.

If we find such a $q$ then we take the following actions:

1. Put
$$A_{s+1} = A_s \cup \{\langle n, x \rangle\}$$

2. Set $f_{s+1}(q) = \text{‘acted’}$.  

3. Reappoint followers for lower priority requirements, i.e. for each $q' > q$
   with $q' < s + 1$ put $x = \langle q', s + 1 \rangle$ to be the follower of $R_{q'}$.

4. Restart lower priority requirements, for each $q' > q$ put
$$f_{s+1}(q') = \text{‘waiting’}.$$  

Finally, assign $x = (s, s + 1)$ to be the follower of $R_s$ and $f_{s+1}(s) = \text{‘waiting’}$.

The verification is virtually the same as in the Friedberg-Muchnic Theorem.
QED

Corollary 65 Every computable partially ordered set embeds into the c.e. degrees $C$. 

43
Proof
Let $\mathbb{P} = (\omega, \leq)$ be a partial order with $\leq$ a computable binary relation on $\omega$. Define $J(p) = \{\langle q, x \rangle \in A : q \leq p\}$ and let $j(p) = \text{deg}(J(p))$. Then
\[ j : \mathbb{P} \to C \]
is an order preserving embedding.
QED

Hmwk 20. (Mon 10-25) Prove there exists a computable partial order $\mathbb{P}_0 = (\omega, \leq_0)$ such that every countable partial order $\mathbb{P}_1$ can be embedded into it, i.e., there exists a 1-1 mapping $j : \mathbb{P}_1 \to \mathbb{P}_0$ such that $p \leq_1 q$ iff $j(p) \leq_0 j(q)$.

It follows from this exercise that every countable partial order embeds into the c.e. degrees.

Hmwk 21. (Wed 10-27) Prove that for every creative set $A$ there exist a set $B$ which is c.e. and disjoint from $A$ but cannot be separated from it by a computable set. Prove that there exists disjoint c.e. sets $A_0$ and $A_1$ which are computably inseparable but not creative.

Theorem 66 (Sacks) Suppose $0 < T C \leq_T 0'$ and $A$ is c.e. Then there exists c.e. sets $A_0$ and $A_1$ such that

1. $A$ is the disjoint union of $A_0$ and $A_1$,
2. $C \not\leq_T A_i$ for $i = 0, 1$, and
3. $A_i$ is of low degree for $i = 0, 1$, i.e., $A_i' \equiv_T 0'$.

Proof
By the limit lemma there exists a computable function $g : \omega \times \omega \to 2$ such that for every $n$
\[ C(n) = \lim_{s \to \infty} g(s, n). \]
To simplify notation let $C_s(n) = g(s, n)$.

Let $A = \{a_s : s \in \omega\}$ be a 1-1 computable enumeration of $A$. If $A$ is finite or even computable the result is trivially true, so we don’t have to worry about that case. We will achieve the splitting of $A$ by simply putting $a_s$ into exactly one of the two sets $A_0$ or $A_1$ at stage $s + 1$. 

44
The lowness of the sets will be achieved by same requirements as in the low simple set proof:

\[ \exists s \quad \{ e \}^{A_i \mid s} \downarrow \rightarrow \{ e \}^{A_i} \downarrow \]

Our new requirements are for each \( e \in \omega \) and \( i = 0, 1 \):

\[ R_{e,i} \quad \{ e \}^{A_i} \neq C \]

which we will write \( R_q = R_{e,i} \) where \( q = 2e + i \). If we meet each of these, then \( C \not\leq_T A_i \) for \( i = 0, 1 \). For each \( q \) we will have two variables \( l_q \) and \( u_q \) which are the length of agreement and the use of some computations. We will use \( u_q \) to satisfy both \( N_q \) and \( R_q \).

We use the notation \( l_q^s \) and \( u_q^s \) to refer to the values of these variables at stage \( s \). At stage \( s = 0 \) put \( A_{i,s} = \emptyset \) and put \( u_q = l_q = 0 \).

**Stage \( s + 1 \).**

Begin by computing the length of agreement \( l_q \) and the usage \( u_q \) for each \( q < s + 1 \):

Suppose \( q = 2e + i \).

(a) If \( \{ e \}^{A_i \mid s}(e) \downarrow \), then:

\[ u_q := \max\{u_q, \text{use}(\{ e \}^{A_i \mid s}(e)) \} \]

(b) Next we adjust the length of agreement. There are two cases:

(1) For all \( x \leq l_q \)

\[ \{ e \}^{A_i \mid s}_s(x) \downarrow = C_s(x) \]

In this case we bump up the usage and increment \( l_q \):

\[ u_q := \max\{u_q, \text{use}(\{ e \}^{A_i \mid s}_s(x)) : x \leq l_q \} \]

\[ l_q := l_q + 1 \]

(2) Not case (1). In this case we do not change \( l_q \) and \( u_q \).

Now we take action. Find the least \( q < s + 1 \) (if any) such that \( a_s < u_q \). If \( q = 2e + i \), then put \( a_s \) into the opposite set, \( A_{1-i} \), i.e.,

\[ A_{1-i,s+1} = A_{1-i,s} \cup \{ a_s \} \]

This means we protect the computations above from being injured.
If no such $q$ exists, then put $a_s$ into $A_0$. This ends the stage and the construction.

Now we verify that the construction works.

**Claim.** For each $q$
1. $R_q$ is met,
2. $\lim_{s \to \infty} l^s_q = L_q < \infty$,
3. $\lim_{s \to \infty} u^s_q = U_q < \infty$, and
4. $N_q$ is met.

**Proof**
In the case of (2) and (3) since our variables are nondecreasing this just means that at some stage they stop growing. The Claim is proved by induction on $q$. So suppose it is true for all $q' < q$ and let $R_q = R_{e,i}$

(1) For contradiction assume that $R_q$ is not met, i.e.,
\[
\{e\}^{A_i} = C.
\]

Subclaim (a). $\lim_{s \to \infty} l^s_q = \infty$.
To see why this is true, note that for any $x$ there will be some stage $s_0$ where $C_s \upharpoonright x = C \upharpoonright x$ for all $s > s_0$ and also $\{e\}^{A_i} \upharpoonright x$ will be same computations as $\{e\}^{A_i}_{s_0,\infty} \upharpoonright x$, i.e., the use of the oracle has settled down. After $s_0$ the variable $l_q$ will be incremented until it is at least $x$, if it isn’t already. This proves subclaim (a).

Now go to a stage $s_0$ such that
1. for all $s > s_0$ and for all $q' < q$ $u^s_{q'} = U_{q'}$ and
2. $a_s > \max\{U_{q'} : q' < q\}$ for all $s > s_0$.

Subclaim (b). If $s > s_0$ is a stage where $l_q$ is incremented then
\[
C(x) = \{e\}^{A_i,s}_{s}(x).
\]
for any $x < l_q$
To see why this is true, note that $u_q$ protects the computation $\{e\}^{A_i,s}_{s}(x)$ from ever changing since $a_s$ is never beneath $u_{q'}$ for any higher priority $q' < q$.
This means that
\[
\{e\}^{A_i,s}_{s}(x) = \{e\}^{A_i}_{s}(x).
\]

46
But we are assuming \( \{e\}^A_i = C \). This proves subclaim (b).

Now we get a contradiction to our assumption that \( C \) is not computable. To compute \( C(x) \) search for a stage \( s > s_0 \) where \( l_q > x \) and it has just been incremented. Then \( C(x) = \{e\}^A_{i,s}(x) \).

This contradiction proves the main Claim part (1) that \( R_q \) is met.

(2) Since \( R_q \) is met there exists \( x \) such that either
(a) \( \{e\}^A_i(x) \uparrow \) or
(b) \( \{e\}^A_i(x) \downarrow \neq C(x) \).

Go to a stage \( s_0 \) such that
1. for all \( s > s_0 \) and for all \( q' < q \) \( u_q^s = U_q' \),
2. \( a_s > \max\{U_q' : q' < q\} \) for all \( s > s_0 \), and
3. \( C_s(x) = C(x) \) for all \( s > s_0 \).

It is impossible that at some stage \( s > s_0 \) where \( l_q > x \) that \( l_q \) is incremented. This is because at \( s \)
\[
\{e\}^A_{i,s}(x) \Downarrow C_s(x)
\]
but \( u_q \) protects the computation \( \{e\}^A_{i,s}(x) \) for the rest of the construction but then
\[
\{e\}^A_i(x) = \{e\}^A_{i,s}(x) = C_s(x) = C(x)
\]
which contradicts the choice of \( x \).

(3) Note that \( u_q \) changes only when either \( l_q \) is incremented or when we see \( \{e\}_{s}^{A_{i,s}}(e) \) converges. Hence if we go to a stage \( s_0 \) such that
1. for all \( s > s_0 \) and for all \( q' < q \) \( u_q^s = U_q' \),
2. \( a_s > \max\{U_q' : q' < q\} \) for all \( s > s_0 \), and
3. \( l_q^s = L_q \) for all \( s > s_0 \)
then \( u_q \) will change at most once more, after which it protects the computation \( \{e\}^A_{i,s}(e) \) from changing and never changes again.

(4) The proof that \( N_q \) is met is the same as in the low simple set argument.

This ends the proof of the Claim and of the Sacks Splitting Theorem.

QED

47
Proposition 67 Suppose $A = A_0 \cup A_1$ is a disjoint union of c.e. sets $A_0$ and $A_1$, then $A \equiv_T A_0 \oplus A_1$.

Proof
Clearly $A = A_0 \cup A_1 \leq_A A_0 \oplus A_1$. To see that $A_i \leq_A A_i$, input $x$ and first ask the oracle if $x \in A_i$. If yes, enumerate $A_0$ and $A_1$ until $x$ shows up.
QED

Corollary 68 (Friedberg Splitting) Every c.e. set which is not computable is the disjoint union of two c.e. sets which are not computable.

Proof
Take $C = A$. Then $A_i \not\leq_T A$ but if either is computable then by the Proposition we get a contradiction.
QED

Corollary 69 For every $c \in D$ if $0 < c < o'$, then there exists $a \in \mathcal{C}$ with $a | c$.

Proof
Let $A = 0'$. By the Theorem $A = A_0 \oplus A_1$ where $C \not\leq_T A_i$ for both $i = 0, 1$. But then at most one of the $A_i$ can be $\leq_T C$, since otherwise

$$0' \equiv_T A_0 \oplus A_1 \leq_T C.$$  

QED

Corollary 70 There exists $a_0, a_1 \in \mathcal{C}$ such that

$$(a_0 \lor a_1)' \neq a_0' \lor a_1'$$

Proof
By the Theorem there exists low c.e. sets $A_i$ such that $A_0 \oplus A_1 \equiv_T 0'$. Hence

$$a_0' \lor a_1' = o' < o'' = (a_0 \lor a_1)'$$

QED

Corollary 71 No c.e. degree is minimal, in fact, beneath any nontrivial c.e. degree is a nontrivial low c.e. degree.
Proof
Given c.e. set \( A \) which is not computable, let \( C = A \) and then we have low c.e. sets \( A_0 \) and \( A_1 \) which split \( A \) and \( A \not\leq_T A_i \). Then for each \( i \) we have that \( 0 <_T A_i <_T A \).
QED

**Hmwk 22.** (Fri 10-29) Define \( f \) is proper iff \( f \) is a partial computable function and both the domain and range of \( f \) are noncomputable subsets of \( \omega \). Prove that for every proper \( f \) there exists proper \( f_0 \) and \( f_1 \) with \( f \) the disjoint union of \( f_0 \) and \( f_1 \).

**Theorem 72** (Lachlan, Yates) There exists a minimal pair of c.e. degrees, i.e. \( a_0, a_1 \in C \setminus \{0\} \) such that the only degree \( b \) with \( b \leq a_0 \) and \( b \leq a_1 \) is \( b = 0 \).

Proof

**Requirements:**
\[
P_{e,i} \quad \psi_e \neq A_i
\]
\[
N_{e_0,e_1} \quad (\{e_0\}^{A_0} = \{e_1\}^{A_1} = B) \rightarrow B \text{ computable.}
\]

**Strategies:**
For \( P_{e,i} \) wait for \( \psi_{e,s}(x) \downarrow = 0 \) for some follower \( x \) and then put \( x \) into \( A_i \).
For \( N_{e_0,e_1} \) restrain agreement to get (a) or (b):
(a) for some \( l < \omega \) we have that \( \{e_0\}^{A_0} \downarrow = \{e_1\}^{A_1} \downarrow \) and either
\[
(\{e_0\}^{A_0}(l) \uparrow \text{ or } \{e_1\}^{A_1}(l) \uparrow) \text{ or } (\{e_0\}^{A_0}(l) \downarrow \neq \{e_1\}^{A_1}(l) \downarrow)
\]
(b) \( \{e_0\}^{A_0} = \{e_1\}^{A_1} = B \) and \( B \) is computable by virtue of our restraining certain computations, that is, we can compute \( B \) by finding stages where we can be sure the approximate computation at that stage is the final one.

**Outcomes:**
For \( P_{e,i} \) the outcomes are either to wait forever or to act at some time. We order them by \( \{ \text{act} < \text{wait} \} \).
For $N_{e_0,e_1}$ the outcomes are either $l < \omega$ where $l$ is the largest length of agreement which we see at a true stage or $\{\infty\}$ if the length of agreement has infinite limit. We use the ordering

$$\infty < \cdots < l + 1 < l < \cdots < 2 < 1 < 0$$

because it is traditional to take limit infimums (rather than limsup) in the outcome tree to determine the truth path.

The outcomes are $\Lambda = \{\text{act, wait}\} \cup \{\infty\} \cup \omega$. The tree of outcomes is $\Lambda^{<\omega}$. At each stage $s$ in the construction we will have computably constructed $f_s \in \Lambda^s$ which is an approximation to the true path, i.e., the eventually correct outcomes.

If $\alpha \in \Lambda^n$ where $n = 2^{(e_0, e_1)}$ then $\alpha$ works on the requirement $N_{e_0,e_1}$. If $\beta \in \Lambda^n$ where $n = 2m + 1$ and $m = 2e + i$, then $\beta$ works on the requirement $P_{e,i}$.

Supplementary variables:

For each such $\beta$ working on a positive requirement we have a restraint variable $R_{\beta} \in \omega$. Also for each such $\beta$ we let

$$F_{\beta} = \{\langle \beta, x \rangle : x \in \omega\}$$

be the followers of $\beta$. These could be any pairwise disjoint family of uniformly computable infinite subsets of $\omega$.

For each $\alpha$ working on a negative requirement we have two variables $l_\alpha$ and $u_\alpha$ (length of agreement and the usage of some computations).

The Construction:

Stage $s = 0$. Put $A_{0,0} = A_{1,0} = \emptyset$ and $f_0 = \langle \rangle$, and put all supplementary variables, $R_{\beta}, l_\alpha, u_\alpha$ equal to zero.

Stage $s + 1$. Given $A_{0,s}, A_{1,s}$, and $f_s \in \Lambda^s$ proceed as follows.

Action:

Look for the least $\beta \subset f_s$ working on a positive requirement $P_{e,i}$ such that

1. $f_s(|\beta|) = \text{’wait’}$ and
2. there exist $x > R_{\beta}$ with $x \in F_{\beta}$ and $x < s$ such that $\psi_{e,s}(x) \downarrow = 0$. 

50
Put the least such $x$ into $A_i$, i.e.,

$$A_{i,s+1} = A_{i,s} \cup \{x\}.$$  

In this case we say that $\beta$ and $P_{e,i}$ acted at stage $s + 1$. If no such $\beta$ exists, then no action is taken.

Update variables:

Define $f_{s+1}|n$ for $n \leq s + 1$ by induction on $n$. At the same time we may update the supplementary variables for each $\gamma \subseteq f_{s+1}$.

**Case $\beta = f_{s+1}|n$ where $\beta$ is working on $P_{e,i}$:**

If $P_{e,i}$ has acted at some stage $\leq s + 1$ then put $f_{s+1}(n) = 'act'$. Otherwise $f_{s+1}(n) = 'wait'.

Define $R_{\beta}$ to be the maximum of the following sets:

1. $\{u_\alpha : \alpha <_{lex} \beta\}$ where $\alpha <_{lex} \beta$ means that there exists $k$ such that $\alpha|k = \beta|k$ and $\alpha(k) < \beta(k)$ in the ordering of outcomes.

2. $\{u_\alpha : \alpha \subseteq \beta$ and $\beta(|\alpha|) \neq \infty\}$.

Remarks. $\beta$ preserves computations of $\alpha$s which are lexicographically to its left because $\alpha$’s want $\beta$’s to their right to respect their computations. $\beta$ also respects computations directly below it except for those which $\beta$ thinks will have an infinite length of agreement.

**Case $\alpha = f_{s+1}|n$ and $\alpha$ is working on $N_{e_0,e_1}$:**

We begin by asking:

Does $\{e_0\}_{A_{i,s+1}}^{A_{i,s+1}}(x) \downarrow = \{e_1\}_{A_{i,s+1}}^{A_{i,s+1}}(x) \downarrow$ for every $x \leq l_\alpha$?

If yes, we put $f_{s+1}(n) = \infty$ and we set:

$$u_\alpha := \max\{u_\alpha, use(\{e_i\}_{A_{i,s+1}}^{A_{i,s+1}}(x)) : x \leq l_\alpha, i = 0, 1\}$$

$$l_\alpha := l_\alpha + 1$$

If no, we put $f_{s+1}(n) = l_\alpha$ and make no changes in the variables.

Remarks. If we see expansion in the length of agreement over what it was when last we set it, we guess optimistically that the length of agreement will expand forever. If we don’t see this expansion, we pessimistically guess we will never see another expansion. (At least on the stages which go thru $\alpha$.)
Verification.

We begin by defining the true path $f \in \Lambda^\omega$. We define $f|n$ by induction on $n$. First let

$$T_n = \{ s > n : f|n \subseteq f_s \}$$

these are the true stages and note that $T_n \subseteq T_{n-1}$. The set $T_n$ is a computable set which (by induction) is infinite. Define $f(n)$ by

$$f(n) = \liminf_{s \in T_n} f_s(n).$$

If $\beta = f|n$ is working on $P_{e,i}$, then $f(n) = 'act'$ if $P_{e,i}$ every acts, and otherwise $f(n) = 'wait'$, meaning we wait forever. In the case $\alpha = f|n$ is working on a negative requirement $f(n)$ will be $\infty$ if there are infinitely many $s \in T_n$ in which the length of agreement $l_\alpha$ has been incremented and otherwise it will be the final value of $l_\alpha$.

**Claim.** For each $n$ the requirement that $f|n$ is working on is met.

**Proof**

**Case** $f|n = \beta$ is working on $P_{e,i}$.

If $f(n) = 'act'$, then for some $x$ we put $x$ into $A_i$ at a stage $s$ where we saw $\psi_e(s(x) \downarrow = 0$. But then $A_i(x) = 1 \neq \psi_e(x)$.

If $f(n) = 'wait'$, let us first prove that $R_\beta$ does not change at any stage $s \geq \min(T_n)$. We first note that for every $s > \min(T_n)$ that it is not true that $f_s <_{\text{lex}} \beta$. Why? Suppose $f_s|k = \beta|k$ and $f_s(k) < \beta(k)$. If $\beta(k) = 'wait'$ and $f_s(k) = 'act'$, then we get a contradiction, since then $\beta$ is not on the true path $f$. In the case of a negative requirement $\alpha = \beta|n$ then $\beta(k) = l < \omega$ (since nothing is to the left of $\infty$), but this would mean that the true path would go to the left of $\beta$. It follows that for every $s \in T_n$ the variables $\{ u_\alpha : \alpha <_{\text{lex}} \beta \}$ will be what they were at the stage $s = \min(T_n)$. Similarly for any $u_\alpha$ with $\alpha \subseteq \beta$ and $\beta(|\alpha|) \neq \infty$ these variables will have also reached their maximum since $u_\alpha$ is only changed when $l_\alpha$ is incremented.

To see that $P_{e,i}$ is met in this case let $R^*_{\beta}$ be this final value of $R_\beta$. Let $x \in F_\beta$ with $x > R^*_{\beta}$. It is not the case that $\psi_e(x) \downarrow = 0$, because if this ever happened then for some large enough stage $s \in T_n$ the worker $\beta$ would have acted (either putting this or some smaller $x$ into $A_i$). Since $x$ is never put into $A_i$ the requirement is met because $\psi_e(x) \neq 0 = A_i(x)$. 

52
Case $f|n = \alpha$ is working on $N_{e_0,e_1}$.

If $f(n) = l$, then for every $s \in T_{n+1}$ the length of agreement was less than $l + 1$, i.e. for some $x \leq l + 1$ it was not true that:

$$\{e_0\}^{A_0,s}(x) \downarrow = \{e_1\}^{A_1,s}(x) \downarrow$$

otherwise we would have incremented $l_\alpha$. It follows that

$$\neg(\{e_0\}^{A_0} = \{e_1\}^{A_1} = B)$$

and so $N_{e_0,e_1}$ is satisfied.

If $f(n) = \infty$, then we claim that $B$ is computable. To see this suppose $s_1 < s_2$ are successive stages in $T_{n+1}$. Note that $\alpha = f_{s_1}|n = f_{s_2}|n$ and $f_{s_1}(n) = f_{s_2}(n) = \infty$. This means that $l_\alpha$ was incremented at each stage $s_i$, say $l - 1$ to $l$ at stage $s_1$ and $l$ to $l + 1$ at stage $s_2$. At stage $s_1$ before any action the two computations agreed:

$$\{e_0\}_{s_1}^{A_0,s_1}|l \downarrow = \{e_1\}_{s_1}^{A_1,s_1}|l \downarrow.$$ 

If $\beta \subseteq f_{s_1}$ is the node which acted at stage $s_1$ (if any), then it must be that $\alpha \subseteq \beta$ and $\beta(n) = \infty$. This action could destroy either the left side or ride side of this agreement but not both, since some $x$ may be put into $A_0$ or $A_1$ but not both. The variable $u_\alpha$ is set to protect the surviving side in subsequent stages. At stages $s$ with $s_1 < s < s_2$ any acting node $\beta$ must be lexicographically to the right of $\alpha \uparrow \infty = f|(n + 1)$, i.e., $f|(n + 1) <_{lex} \beta$. But this means that $R_\beta \geq u_\alpha$ and so the action at stage $s$ cannot damage the surviving side. At stage $s_2$ we increment $l$ to $l + 1$ which means that the destroyed side must have come back and equaled the surviving side. This means that for each $s \in T_{n+1}$:

$$\{e_0\}_{s}^{A_0}|l_{s}^{s} = \{e_0\}_{s}^{A_0,s}|l_{s}^{s}$$

i.e., the final computation is the computation we see at this stage. Hence to compute $B(x)$ search for a stage $s \in T_{n+1}$ such that $x < l_\alpha$ and then $B(x) = \{e_0\}_{s}^{A_0,s}(x)$. It follows that $B$ is computable. This proves the Claim and the minimal pair theorem.

QED

Hmwk 23. (Fri Nov 5) Put the low simple non-auto reducible set construction on a tree of outcomes. Prove the construction works. Show that there is no injury on the true path.
Theorem 73 (Friedberg, Enumeration without repetition) There exists an c.e. set $U$ such that

1. $\{U_e : e \in \omega\}$ is the set of all c.e. sets and
2. $U_{e_1} \neq U_{e_2}$ for all $e_1 \neq e_2$

Proof

We will first construct an c.e. set $V$ and then modify it to get $U$. The requirements are:

$R_e \quad \forall \hat{e} < e \ (W_{\hat{e}} \neq W_e) \to W_e = V_x$ for some unique $x$.

The strategy for meeting this requirement is to appoint a follower $x$. As long as it looks like $\forall \hat{e} < e \ (W_{\hat{e}}|x \neq W_e|x)$ keep enumerating $W_e$ into $V_x$. Otherwise make it a disloyal follower and put it into the garbage. What do we do with $V_x$ when $x$ is a disloyal follower? We make it into an initial segment.

Definition 74 $A \subseteq \omega$ is an initial segment iff $A = \emptyset$ or $A = \omega$ or there exists $n < \omega$ such that $A = [0, n] = \{i < \omega : 0 \leq i \leq n\}$.

So our modified requirement is:

$R_e \quad \text{If } \forall \hat{e} < e \ (W_{\hat{e}} \neq W_e) \text{ and } W_e \text{ is not an initial segment, then } W_e = V_x$ for some unique $x$.

At stage $s + 1$ in our construction we have the following sets:

1. $F_s$ the followers
2. a 1-1 mapping from $F_s$ to $\omega$ which tells us that $x$ is the follower of $e$, say $f_s(x) = e$
3. $D_s$ the disloyal former followers
4. $(V_{x,s} : x \in F_s \cup D_s)$
5. a nondecreasing variable $g_s$ keeping track of last initial segment assigned to a disloyal follower.

The sets $F_s$ and $D_s$ will be disjoint finite sets whose union is an initial segment.
Construction

**Stage** $s + 1$

Let $s = \langle e, ? \rangle$. (So we visit each $e$ infinitely often.)

If no follower is assigned to $R_e$, let $x = \min(F_s \cup D_s)$ and assign $x$ to be the follower of $R_e$. Put $F_{s+1} = F_s \cup \{x\}$ and end the stage.

If $x$ is the follower of $R_e$ and

1. $\forall \hat{e} < e$ \hspace{1cm} $W_{\hat{e}, s+1} \cap [0, x] \neq (W_{e, s+1}) \cap [0, x]$

2. $W_{e, s+1} \cap [0, x]$ is not an initial segment

then put $V_{x, s+1} = V_{x, s} \cup W_{e, s+1}$ and end the stage. Actually in this case $V_{x, s} \subseteq W_{e, s}$ so we could have said put $V_{x, s+1} = W_{e, s+1}$.

If $x$ is the follower of $R_e$ and either of those two conditions fails then

1. change $x$ into a disloyal follower, i.e., $F_{s+1} = F_s \setminus \{x\}$ and $D_{s+1} = D_s \cup \{x\}$,

2. let $g_{s+1}$ be the minimum $g > g_s$ such that $V_{e, s} \subseteq [0, g]$, and

3. permanently assign $V_x$ to be $[0, g_{s+1}]$, i.e., set $V_{x, s+1} = [0, g_{s+1}]$ and never change $V_x$ again.

End the stage.

**Verification**

**Claim.** The following are equivalent for any $e$:

1. For each $e$ if $W_e$ is not an initial segment of $\omega$ and $W_e \neq W_{\hat{e}}$ for each $\hat{e} < e$.

2. $R_e$ obtains a permanent follower $x$ and hence $V_x = W_e$.

**Proof**

Suppose condition 2 holds. Then $R_e$ obtains a permanent follower $x$. Then for all stages $s + 1$ after $x$ is appointed and for which $s = \langle e, ? \rangle$, we have that $W_{e, s} \cap [0, x]$ is not an initial segment and $W_{e, s} \cap [0, x] \neq W_{\hat{e}, s} \cap [0, x]$ for each $\hat{e} < e$. Condition (1) follows since there are infinitely many such stages.
Suppose that condition 1 holds. Choose \( y \) so that \( W_e \cap [0, y] \) is not an initial segment and
\[
W_e \cap [0, y] \neq W_{\hat{e}} \cap [0, y]
\]
for every \( \hat{e} < e \). Go to some stage \( s_0 \) where
\[
W_{e,s_0} \cap [0, y] = W_e \cap [0, y]
\]
and
\[
W_{\hat{e},s_0} \cap [0, y] = W_{\hat{e}} \cap [0, y]
\]
for every \( \hat{e} < e \). If \( R_e \) has no permanent follower then infinitely many followers are appointed to it. Hence some follower \( x > y \) will be appointed after stage \( s_0 \). But such a follower will always remain loyal.
QED

Let \( D = \bigcup_{s \in \omega} D_s \) be the set of disloyal followers. Then \( \overline{D} \) is the set of permanent followers.

**Claim.**

1. \( \{ V_x : x \in \overline{D} \} \) is the set of c.e. sets which are not initial segments.
2. There exist a computable set \( G \) such that
\[
\{ [0, n] : n \in G \} = \{ V_x : x \in D \}.
\]
3. \( V_x \neq V_{x'} \) unless \( x = x' \).

**Proof**

Part (1) follows from the first Claim.

For Part (2), since the sequence \( g_s \) is non-decreasing we see that
\[
G = \{ g_s : s \in \omega \}
\]
is computable.

For Part (3) note that there are two types of \( V_x \). If \( x \) is a permanent follower of some \( R_e \) and then \( V_x = W_e \) where \( W_e \) is not an initial segment and \( W_e \) is distinct from each \( W_{\hat{e}} \). Or \( x \) is a disloyal follower at some stage \( s + 1 \) and then \( V_x = [0, g_{s+1}] \). Since the sequence \( g_s \) is bumped up each time it is used we see that the \( V_x \) for disloyal followers are distinct finite initial segments. This proves Claim.
Let us show how to modify $V$ to $U$ to prove Friedberg’s enumeration without repetition theorem. Note that $V$ uniquely enumerates every c.e. set except $\omega$, $\emptyset$, and the finite initial segments of the form $[0, n]$ where $n \notin G$. Let $\{x_n : 1 < n < \omega\}$ be a 1-1 computable enumeration of $G$. Now define $U$ by $U_0 = \omega$, $U_2 = \emptyset$, $U_{2n} = [0, x_n]$ for $n > 1$, and $U_{2n+1} = V_n$.

QED

Hmwk 24. (Mon Nov 8)
(a) Prove there exists $V$ c.e. such that $\{V_e : e \in \omega\} = \text{set of c.e. non-simple sets.}$
(b) Prove there exists $U$ c.e. such that $\{U_e : e \in \omega\} = \text{set of c.e. non-simple sets and } U_{e_1} \neq U_{e_2} \text{ unless } e_1 = e_2.$

Definition 75 Coding finite sets. For $D \subseteq \omega$ let $x = \sum_{n \in D} 2^n$. Write $D_x = D$.

Definition 76 $(D_x : x \in R)$ is a strong array iff $R$ is an infinite computable set and for every $x, y \in R$ we have $D_x \cap D_y = \emptyset$ whenever $x \neq y$.

Definition 77 A set $A \subseteq \omega$ is hypersimple iff $A$ is c.e., $A$ is infinite, and for every strong array $(D_x : x \in R)$ there exists $x \in R$ such that $D_x \subseteq A$.

Proposition 78 (Post)
(1) Hypersimple implies simple.
(2) There is a simple set which is not hypersimple.
(3) There is a hypersimple set.

Proof
(1) If $A$ is not simple, then there exists an infinite computable set $R \subseteq \overline{A}$. Then $\{D_{2e} : x \in R\}$ witnesses that $A$ is not hypersimple.
(2) In Post’s original construction of a simple set $A$ (see Theorem 35) we constructed a simple set $A$ by waiting until there was some $x \in W_{e,s}$ with $x > 2e$ and $W_{e,s} \cap A_s = \emptyset$ and then putting $x$ into $A$. The reason that $\overline{A}$ was infinite was because for every $e$ we had that $|[0, 2e] \cap A| \leq e$. This means that for every $a$ we have that $[a, 4a] \cap \overline{A} \neq \emptyset$.
because \([a, 4a]\) is 3/4 of the interval \([0, 4a]\). So define \(a_0 = 5\) and \(a_{n+1} = 4a_n + 1\). Take \(x_n\) so that \(D_{x_n} = [a_n, 4a_n]\) and note that \(D_{x_n} \cap \overline{A} \neq \emptyset\) for each \(n\) so the computable set \(R = \{x_n : n < \omega\}\) witnesses that \(A\) is not hypersimple.

(3) This is a consequence of the following proposition, although originally Post gave a construction similar to his construction of a simple set.

QED

**Proposition 79** (Dekker) Deficiency sets are hypersimple.

Proof

See Theorem 37. Suppose that \(A = \{a_s : s \in \omega\}\) is a 1-1 computable enumeration of \(A\) and \(A\) is not computable. Define

\[
D = \{s : \exists t > s \ a_t < a_s\}.
\]

As we saw before \(A \equiv_T D\) and \(D\) is simple. A similar proof will show that \(D\) is hypersimple.

Suppose for contradiction that there exists a strong array \((D_x : x \in R)\) such that \(D_x \cap \overline{D} \neq \emptyset\) for every \(x \in R\).

Now we get a contradiction by showing that \(A\) is computable.

Input \(u\). Find an \(x \in R\) such that

\[
u < \min\{a_s : s \in D_x\}.
\]

Such an \(x\) exists, since \(a_s\) is a 1-1 enumeration and the \(D_x\) are pairwise disjoint. But now at least one of \(t \in D_x\) is not deficient, so for all \(s > t\) we have \(a_s > a_t\). Hence \(u \in A\) iff \(u = a_s\) for some \(s \leq \max D_x\).

QED

**Hmwk 25.** (Wed Nov 10) Define \(A\) to be bdd-hypersimple iff \(A\) is c.e., \(A\) is infinite, and for every strong array \((D_x : x \in R)\) such that there exists \(N < \omega\) such that \(|D_x| \leq N\) for all \(x \in R\), there exists \(x \in R\) such that \(D_x \subseteq A\). Prove that bdd-hypersimple is equivalent to simple.

**Definition 80** For any set \(A \subseteq \omega\) such that \(\overline{A}\) is infinite define \(g_A \in \omega^\omega\) by \(g_A(n)\) is the \((n+1)\text{th}\) element of \(\overline{A}\), i.e.,

\[
\overline{A} = \{g_A(0) < g_A(1) < \cdots < g_A(n) < \cdots\}.
\]
Proposition 81  For any c.e. set $A$ with $\overline{A}$ infinite the following are equivalent:

1. $A$ is hypersimple.
2. For any computable increasing sequence $n_k < n_{k+1}$ there are infinitely many $k$ with $[n_k, n_{k+1}) \subseteq A$.
3. For any computable $f \in \omega^\omega$ there are infinitely many $k$ such that $f(k) < g_A(k)$.

Proof

(1) $\rightarrow$ (2). This is clear since if $D_{x_k} = [n_k, n_{k+1})$, then $R = \{x_k : k < \omega\}$ is a strong array. There are infinitely many since $R(l) = \{x_k : k > l\}$ is a strong array for any $l$.

(2) $\rightarrow$ (3). Given a computable $f$ construct a computable sequence $n_{k+1} > n_k$ with the property that $f(n_{k+1}) < n_{k+1}$ for each $k$. For any $k$ such that $[n_k, n_{k+1}) \subseteq A$ note that $\overline{A} \cap [0, n_{k+1}) \subseteq [0, n_k)$ and so $g_A(n_{k+1}) = (n_k + 1)^{th}$ element of $\overline{A}$ must be greater than $n_{k+1}$. Hence $f(n_{k+1}) < g_A(n_{k+1})$.

(3) $\rightarrow$ (1). Suppose $A$ is not hypersimple and hence there exists a strong array $(D_x : x \in R)$ such that $D_x \cap \overline{A} \neq \emptyset$ for all $x \in A$. Let $\{x_n : n \in \omega\}$ be a 1-1 computable enumeration of $R$ and define

$$f(n) = 1 + \max(\cup_{m \leq n} D_{x_m})$$

Then $|\overline{A} \cap [0, f(n)]| > n$ and so $f$ dominates $g_A$.

QED

Hmwk 26. (Fri 11-12) Prove that for every c.e. set $A \subseteq \omega$ if $\overline{A}$ is infinite, then there exists a hypersimple set $B \supseteq A$.

Consider propositional logic with the set of atomic letters

$$\{P_n : n \in \omega\}.$$ 

For any propositional sentence $\psi$ and subset $A \subseteq \omega$ define

$$A \models \psi$$

inductively by

$$A \models P_n \text{ iff } n \in A$$
\[ A \models \neg \psi \text{ iff not } A \models \psi \]
\[ A \models (\psi \lor \theta) \text{ iff } (A \models \psi \text{ or } A \models \theta) \]

and so forth for the other logical symbols.

By coding symbols as elements of \( \omega \) and thinking of sentences as strings of symbols or finite sequences of elements of \( \omega \), we identify the set of propositional sentences with a computable subset of \( \omega \), \( \text{SENT} \). The details of this coding are left to the reader.

The following notion is known as truth-table (tt) reducibility.

**Definition 82** \( A \leq_{tt} B \) iff there exists a computable sequence

\[ (\theta_n \in \text{SENT} : n \in \omega) \]

such that for all \( n \in \omega \)

\[ n \in A \text{ iff } B \models \theta_n \]

Note: It is easy to see that \( A \leq_{tt} C \) and \( B \leq_{tt} C \) implies \( (A \cap B) \leq_{tt} C \) and \( \overline{A} \leq_{tt} C \). Hence the family of sets which are truth-table reducible to \( C \) is closed under finite boolean combinations. It is easy to see that \( \leq_{m} \)-reducible is stronger than \( \leq_{tt} \), and \( \leq_{tt} \) is stronger than \( \leq_{T} \).

**Proposition 83** (Nerode) The following are equivalent:

1. \( A \leq_{tt} B \).

2. There exist \( e \) with the property that

\[ \forall X \forall x \{e\}^X(x) \downarrow \]

and \( \{e\}^B = A \).

3. There exists \( e \) and \( f \in \omega^\omega \) computable such that

\[ \forall x \{e\}^{f(x)}(x) \downarrow \]

and \( \{e\}^B = A \).
Proof

(1) → (2). Given \((\theta_n : n \in \omega)\) witnessing that \(A \leq_t B\), it is easy to construct an oracle machine \(e\) such that for any input \(x\) and oracle \(X\) that \(\{e\}^X(x) \downarrow = 1\), if \(X \models \theta_x\) and \(\{e\}^X(x) \downarrow = 0\), if \(X \models \neg \theta_x\).

(2) → (3). We show that the same \(e\) works. Input \(x\) and let
\[
T_x = \{ \sigma \in 2^{\omega} : \{e\}_R^\sigma(x) \downarrow \}
\]
The trees \(T_x\) are uniformly computable in \(x\). By Konig’s tree lemma, since \(T_x\) has no infinite branch, it is finite. Therefore we can compute the least \(n\) such that for all \(\sigma \in 2^n\) we have that \(\sigma /\in T_x\). Put \(f(x) = n\).

(3) → (1). Input \(x\). Compute a use bound \(u_x\) so that for every possible computation \(\{e\}_{f(x)}(x)\) the computation only asks about \(i < u_x\). (Since it takes at least one step to ask the oracle anything there are at most \(2^{f(x)}\) such simulations.)

Now define
\[
t_x = \{ R \subseteq [0, u_x] : \{e\}_R^B(x) \downarrow = 1 \}.
\]
Define
\[
\theta_x = \forall_{R \in t_x} ( \forall_{i \in R} P_i \land \forall_{i \in [0, u_x] \setminus R} \neg P_i )
\]
Then for any \(x \in \omega\) we have that
\[
x \in A \text{ iff } \{e\}_{f(x)}^B(x) \downarrow = 1 \text{ iff } B \cap [0, u_x] = R \in t_x \text{ iff } B \models \theta_x.
\]
QED

Proposition 84 (Post)

1. If \(A\) is simple, then \(A <_m K\).
2. If \(A\) is hypersimple, then \(A <_t K\).
3. There exists a simple \(A\) with \(A \equiv_t K\).

Proof

(1) If \(K \leq_m A\) then \(A\) is creative and hence not simple. (See Theorem 33.)

(2) Since every c.e. set is many-one reducible to \(K\) it is enough to see that \(K \leq_t A\) implies \(A\) is not hypersimple.

Claim. Let \(\Gamma = \{ P_n : n \in A\} \). Then there exists a computable list \((\rho_n : n < \omega)\) of propositional sentences such that for every \(n\)
1. \( A \models \rho_n \) and
2. \( \Gamma \cup \{ \rho_m : m < n \} \not\models \rho_n \).

Proof

Since \( \overline{K} \leq_{tt} A \) there exists a computable function \( \theta : \omega \to \text{SENT} \) such that \( n \in \overline{K} \) iff \( A \models \theta(n) \).

Now we effectively construct \( \rho_n \) as follows. Let

\[
\Sigma_n = \{ \rho : \Gamma \cup \{ \rho_m : m < n \} \vdash \rho \}.
\]

Note that \( \Sigma_n \) is computably enumerable as a subset of \( \text{SENT} \). Also \( A \models \theta \) for every \( \theta \in \Sigma_n \). It follows that \( \theta^{-1}(\Sigma_n) \subseteq \overline{K} \) is c.e. By the S-n-m Theorem there exists a computable function \( f \) such that

\[
W_{f(n)} = \theta^{-1}(\Sigma_n)
\]

and by the proof that \( K \) is creative we have that

\[
f(n) \in \overline{K \cup \theta^{-1}(\Sigma_n)}.
\]

Take \( \rho_n = \theta(f(n)) \).

QED

Let \( S_k \) be that set of all \( n \) such that the propositional letter \( P_n \) occurs in the sentence \( \rho_k \), i.e., \( S_k \) is the support of \( \rho_k \).

Claim. For any \( n \) let

\[
m = \max \left( \bigcup \{ S_k : k \leq 2^{2n+1} + 1 \} \right)
\]

then \( \overline{A} \cap [n, m) \neq \emptyset \).

Proof

Suppose not and assume that \( [n, m) \subseteq A \). Let \( \rho_k^* \) be obtained from \( \rho_k \) by replacing all propositional letters \( P_i \) for \( n < i < m \) by the letter \( P_n \). Note that \( \Gamma \vdash P_i \) for all these \( i \) and hence \( \Gamma \vdash \rho_k^* \equiv \rho_k \) for every \( k \leq 2^{2n+1} + 1 \). But there are at most \( 2^{2n+1} \) logically inequivalent propositional sentences with atomic letters \( P_i \) for \( i \leq n \) and so for some \( k < l \) we have that \( \rho_k^* \equiv \rho_l^* \). But this is a contradiction since then

\[
\Gamma \vdash \rho_i \equiv \rho_j.
\]
Now it is an easy matter to construct a computable sequence $n_k < n_{k+1}$ so that $\overline{A} \cap [n_k, n_{k+1}) \neq \emptyset$ for each $k$. Hence $A$ is not hypersimple.

(3) Let $B$ be any simple set which is not hypersimple. By Proposition 81 there exists a computable increasing sequence $(n_k : k < \omega)$ such that for all $k$ we have that $\overline{B} \cap [n_k, n_{k+1}) \neq \emptyset$. Now let

$$A = B \cup \bigcup_{k \in K} [n_k, n_{k+1})$$

$A$ is simple because it is a superset of the simple set $B$. $\overline{A}$ is infinite because for each $k \in K$ we have $\overline{A} \cap [n_k, n_{k+1}) \neq \emptyset$. We have that $K \leq_t A$ because

$$k \in K \text{ iff } A \models \bigwedge_{n_k \leq i < n_{k+1}} P_i$$

QED

Definition 85 $V$ is a weak array iff $V$ is c.e. and $V_x \cap V_y = \emptyset$ whenever $x \neq y$. As usual, $V_x = \{y : \langle x, y \rangle \in V\}$.

Definition 86 $A \subseteq \omega$ is hyperhypersimple iff $A$ is re, $\overline{A}$ is infinite, and for every weak array $V$ there exists $x$ with $V_x \subseteq A$.

Proposition 87 For any $A \subseteq \omega$ for which $A$ is c.e. and $\overline{A}$ is infinite the following are equivalent:

1. $A$ is hyperhypersimple

2. for every infinite c.e. set $B$ such that $W_x \cap W_y = \emptyset$ for all distinct $x, y \in B$ there exists $x \in B$ with $W_x \subseteq A$

3. for every weak array $V$ there exists an infinite computable set $R$ such that $V_x \subseteq A$ for all $x \in R$

4. for every weak array $V$ such that $V_x$ is finite for all $x$ there exists $x$ such that $V_x \subseteq A$
Proof

(1) iff (2) is true because the two types of arrays are the same.

(1) \rightarrow (3). The sequence \((R_n = \{\langle n, m \rangle : m \in \omega \} : n < \omega)\) is a uniformly computable partition of \(\omega\) into infinite pieces. Take

\[U_n = \bigcup_{e \in R_n} V_e\]

Then \(U\) is weak array and so there exists \(n\) with \(U_n \subseteq A\).

(4) \rightarrow (1). Given a weak array \(V\) such that \(V_e \cap A \neq \emptyset\) for all \(e\) we find another weak array \(V^*\) such that \(V^*_e\) finite and \(V^*_e \cap \overline{A} \neq \emptyset\) for all \(e\). For each \(s\) define \(V^*_e,s = V_{e,s_0+1}\) where \(s_0\) is the largest \(t \leq s\) such that \(V_{e,t} \subseteq A_s\).

QED

Hmwk 27. (Mon 11-15) Prove

(a) If \(A\) is simple and \(B\) is simple, then \(A \cap B\) is simple.

(b) If \(A\) is hypersimple and \(B\) is hypersimple, then \(A \cap B\) is hypersimple.

(b) If \(A\) is hyperhypersimple and \(B\) is hyperhypersimple, then \(A \cap B\) is hyperhypersimple.

Example 88 There exists a hypersimple set \(A\) which is not hyperhypersimple.

Proof

Let \(B \subseteq \omega\) be any hypersimple set. Define \(A \subseteq \omega\) by

\[A = \{\langle n, m \rangle : n \in B \text{ or } n \leq m\}.

\(A\) is not hyperhypersimple since each of the sets \(V_n = \{\langle m, n \rangle : m \in \omega\}\) meets \(\overline{A}\). To see that \(A\) is hypersimple suppose we are given a strong array \((D_n : n \in R)\). Let \(p(\langle m, n \rangle) = m\) be projection to the first coordinate. We can find an infinite computable subset \(S \subseteq R\) such that \((\pi(D_x \cap Q) : x \in S)\) are pairwise disjoint where \(Q = \{\langle n, m \rangle : m < n < \omega\}\). Since \(B\) is hypersimple, there exists \(x \in S\) with \(\pi(D_x \cap Q) \subseteq B\) and hence \(D_x \subseteq A\).

QED

Example 89 Dekker deficiency sets are never hyperhypersimple.

Proof
Let \( A = \{ a_s : s \in \omega \} \) be a one-one computable enumeration of a non computable set \( A \). And \( D = \{ s : \exists t > s \ a_t < a_s \} \). We construct a weak array \( V \) to meet the requirements:

\[
R_x \quad V_x \cap \overline{D} \neq \emptyset
\]

Stage \( s+1 \)

Step (a). For any \( x \leq s \) if \( R_x \) has a follower \( t \) such that \( a_s < a_t \) then unappoint \( t \) so that now \( R_x \) has no follower.

Step (b). For the least \( x \) for which \( R_x \) has no follower, appoint \( s \) the follower of \( R_x \) and put \( V_{x,s+1} = V_{x,s} \cup \{ s \} \).

This ends the stage and the construction. Note that \( V \) is a weak array.

**Claim.** Each \( R_x \) obtains a permanent follower \( s \) and for this \( s \) we have \( s \in V_x \cap \overline{D} \).

**Proof**

This is by induction on \( x \). So after some sufficiently large stage \( s_0 \) no \( y < x \) is appointed a new follower. Suppose for contradiction that \( R_x \) is appointed a new follower at stages \( s_1, s_2, \ldots \) where \( s_0 < s_1 < s_2 < \cdots \). Note that since higher priority requirements don’t get new followers after \( s_0 \) each time \( R_x \) losses its follower it acquires the stage itself as its new follower. But this means that

\[
a_{s_1} > a_{s_2} > a_{s_3} > \cdots
\]

which is a contradiction.

QED

**Definition 90** \( A \subset^* B \) iff \( B \setminus A \) is finite.
\[ A =^* B \text{ iff } A \subset^* B \text{ and } B \subset^* A \]
\[ \forall^\infty \text{ means ‘for all but finitely many’} \]
\[ \exists^\infty \text{ means ‘exists infinitely many’} \]

**Definition 91** \( M \subseteq \omega \) is maximal iff \( M \) is c.e., \( \overline{M} \) is infinite, and for every \( A \) c.e. if \( M \subseteq A \) then \( M =^* A \) or \( A =^* \omega \).

**Proposition 92** Maximal implies hyperhypersimple.

Suppose \( V \) is a weak array such that \( V_e \cap \overline{A} \neq \emptyset \) for all \( e \). Define

\[
B = A \cup \bigcup_{e < \omega} V_{2e}
\]
then $A \neq^* B$ and $B \neq^* \omega$, so $A$ is not maximal.
QED

**Theorem 93** (Friedberg) Maximal sets exist.

**Proof**

We will construct the maximal set $M$ as follows. We use the notation $p_n$ for the $n^{th}$ element of the complement of $M$, i.e.,

$$\overline{M} = \{p_0 < p_1 < p_2 < \cdots\}$$

Are requirements are

$$R_e \quad (\forall^\infty n \ p_n \in W_e) \quad \text{or} \quad (\forall^\infty n \ p_n \notin W_e)$$

This guarantees that $M \cup W_e =^* \omega$ or $M \cup W_e =^* M$.

At stage $s$ given $M_s$ we let

$$\overline{M}_s = \{p_{0,s} < p_{1,s} < p_{2,s} < \cdots\}$$

The idea of this proof is called moving markers. We think of a marker labeled $n$ with position $p_{n,s}$. As we slide the marker upward we put the uncovered numbers into $M_s$. In order to get $\overline{M}$ infinite we want each marker to eventually stop moving.

**Definition 94** $\sigma \in 2^n$ is the $n$-state of $x$ at stage $s$ iff

$$\text{for all } e < n \quad \sigma(e) = \begin{cases} 1 & \text{if } x \in W_{e,s} \\ 0 & \text{if } x \notin W_{e,s} \end{cases}$$

Two easy facts about the $n$-state are the following:

1. Suppose $s_1 \leq s_2$, $\sigma_1 \in 2^n$ is the $n$-state of $x$ at stage $s_1$, and $\sigma_2 \in 2^n$ is the $n$-state of $x$ at stage $s_2$,
   then $\sigma_1 \leq_{\text{lex}} \sigma_2$.

2. For fixed $n$ and $x$ there is $\sigma \in 2^n$ such that $\sigma$ is the $n$-state of $x$ for all but finitely many stages $s$. We call this the final $n$-state of $x$.

Our strategy can be summarized simply as ‘maximize the lexicographic order of the $n$-state of $p_n$’.
Stage $s + 1$.

Find the least $n$ (if any) such that there exists $m$ with $n < m < s$ such that
if $\sigma \in 2^n$ is the $n$ state of $p_{n,s}$ and
$\tau \in 2^n$ is the $n$ state of $p_{m,s}$, then $\sigma <_{\text{lex}} \tau$.

For the least such $n$ find the least $m$ and shift the marker $n$ to $m$:

Put $p_{n+i,s+1} = p_{m+i,s}$ for all $i < \omega$. Equivalently put

$$M_{s+1} = M_s \cup \{p_{j,s} : n \leq j < m\}$$

Otherwise as usual if there are no such $n, m$ just go to the next stage with everything unchanged.

This ends the stage and the construction.

**Claim.** The markers eventually stop moving, i.e.,

$$\lim_{s \to \infty} p_{n,s} = p_n < \infty$$

**Proof**
This is proved by induction on $n$. Note that the only way the marker $n$ moves is either that it is bumped up by some marker $m < n$ or it moves to a higher $n$-state. So consider some stage $s_0$ so that no marker $m < n$ moves after stage $s_0$. But it is impossible for $p_n$ to change infinitely many times after this since its $n$-state would have to increase lexicographically infinitely many times. (Note that in between moves its $n$-state might also change without the marker moving but it can only increase if it doesn’t move.)

QED

**Claim.** For each $n$ there exists $\tau \in 2^n$ such that

$$\forall \infty m \quad \tau = \text{the final } n\text{-state of } p_m.$$

**Proof**
Suppose not. Then there exists distinct $\tau_1, \tau_2 \in 2^n$ such that

$\exists \infty m \quad \tau_1 = \text{the final } n\text{-state of } p_m$ and

$\exists \infty m \quad \tau_2 = \text{the final } n\text{-state of } p_m$.

Suppose $\tau_1 <_{\text{lex}} \tau_2$. Then we can choose $m_1, m_2$ with $n < m_1 < m_2$ and the final $n$-state of $p_{m_i}$ is $\tau_i$. This is a contradiction, since for some large enough stage $s_0 > m_2$ the markers $p_j$ for $j \leq m_2$ have stopped moving and their final $n$-states are their states at stage $s_0$. But by the construction some marker $\leq p_{m_1}$ must move.
QED

This final claim proves the Theorem, since if \( n = e + 1 \) we have that
\[
\tau(e) = 1 \text{ implies } \forall^\infty m \ p_m \in W_e
\]
and
\[
\tau(e) = 0 \text{ implies } \forall^\infty m \ p_m \notin W_e
\]
QED

Example 95  There exists a hyperhypersimple set which is not maximal.

Proof
First we note that it easy to get \( M_1 \) and \( M_2 \) maximal so that \( M_1 \neq^* M_2 \).
Take any maximal set \( M \) and let \( R \subseteq M \) to be an infinite computable subset.
Let \( \pi : \omega \rightarrow \omega \) be a computable bijection which takes \( R \) to \( \bar{R} \).
Let \( M_1 = M \) and let \( M_2 = \pi(M_1) \).
Now let \( A = M_1 \cap M_2 \). Then \( A \) is hyperhypersimple (see exercise) but
not maximal since \( A \subseteq M_1 \subseteq \omega \) and \( A \neq^* M_1 \) and \( M_1 \neq^* \omega \).
QED

Remark. Yates noted that we can add to the maximal set construction
an extra ‘kick’ to the \( p_e \) marker to ensure that \( \{e\}(e) \downarrow \) iff \( \{e\}_{pe}(e) \).
Then the maximal set constructed will be Turing equivalent to \( K \).

Hmwk 28.  (Wed 11-17) Suppose \( A = \{a_n : n < \omega\} \) is a 1-1 computable
enumeration of a hyperhypersimple set \( A \). Let \( B = \{a_{a_n} : n < \omega\} \).
Prove that \( B \) is hyperhypersimple but not maximal.

Hmwk 29.  (Fri 11-19) An c.e. set \( A \subseteq \omega \) is simple in \( R \) where \( R \) is an infinite
computable set iff \( \overline{A} \cap R \) is infinite but contains no infinite c.e. subset. Is
every c.e. set which is not computable simple in some infinite computable
set? Hint: Consider a Friedberg splitting of a Maximal set.

Definition 96  The lattice of c.e. sets is \( \mathcal{E} = (\text{c.e.sets}, \subseteq) \). A subset \( X \subseteq \mathcal{E} \)
is definable iff there is a first order formula \( \theta(v) \) in the language of \( \subseteq \) such that
\[
X = \{ A \in \mathcal{E} : \mathcal{E} \models \theta(A) \}.
\]
Similarly for \( X \subseteq \mathcal{E}^2 \) or \( X \subseteq \mathcal{E}^3 \).

Example 97  The following are definable in \( \mathcal{E} \).

1. \( \{(A, B, C) \in \mathcal{E}^3 : A \cup B = C\} \)
2. \( \{(A, B, C) \in E^3 : A \cap B = C\} \)
3. \( \{\emptyset\} \)
4. \( \{\omega\} \)
5. computable sets
   \( A \) is computable iff \( E \models \exists B \ B \cap A = \emptyset \) and \( B \cup A = \omega \)
6. c.e. but not computable sets
7. infinite c.e. sets
   \( A \) is infinite c.e. iff \( E \models \exists B \ B \subseteq A \) and \( B \) is not computable
8. finite sets
9. cofinite
10. simple sets
11. maximal sets

**Definition 98** \( \pi \) is an automorphism of \( E \) iff \( \pi : E \to E \) is a bijection such that for every \( A, B \in E \)

\[
A \subseteq B \text{ iff } \pi(A) \subseteq \pi(B).
\]

Note that for any first-order formula \( \theta(v_1, \ldots, v_n) \) in the language of \( E \), i.e., \( \subseteq \), that for any \( \pi \in \text{aut}(E) \) and \( A_1, \ldots, A_n \in E \) we have that

\[
E \models \theta(A_1, \ldots, A_n) \text{ iff } E \models \theta(\pi(A_1), \ldots, \pi(A_n))
\]

Hence definable sets are closed under automorphisms.

**Example 99** If \( A \in E \), then \( \{A\} \) is definable in \( E \) iff \( A = \emptyset \) or \( A = \omega \).

**Proof**
If \( A \) is neither \( \emptyset \) or \( \omega \), then we can choose \( n, m < \omega \) such that \( n \in A \) and \( m \notin A \). Let \( \pi : \omega \to \omega \) be the identity except \( \pi(n) = m \) and \( \pi(m) = n \). Define \( \pi : P(\omega) \to P(\omega) \) by \( \pi(A) = \{\pi(n) : n \in A\} \). Then since \( \pi \) is computable it is clear that \( \pi \in \text{aut}(E) \). But since

\[
\pi(A) = (A \setminus \{n\}) \cup \{m\}
\]
we see that \( \{A\} \) is not closed under automorphisms and hence cannot be definable.

QED

**Proposition 100**

1. For every \( \pi \in \text{aut}(\mathcal{E}) \) there exists a bijection \( \hat{\pi} \) of \( \omega \) such that \( \pi(\{n\}) = \{\hat{\pi}(n) : n \in A\} \).

2. Not every bijection \( \pi : \omega \to \omega \) induces an automorphism of \( \mathcal{E} \).

3. There are continuum many bijections \( \pi : \omega \to \omega \) which induce an automorphism of \( \mathcal{E} \).

**Proof**

1. It is easy to see that the set of singletons

\[
\{\{n\} : n \in \omega\} \subseteq \mathcal{E}
\]

is definable in \( \mathcal{E} \). Hence any automorphism \( \pi : \mathcal{E} \to \mathcal{E} \) must permute the singletons. Define \( \hat{\pi}(n) \) so that \( \pi(\{n\}) = \{\hat{\pi}(n)\} \). But now for every \( n \in \omega \)

\[
n \in A \iff \{n\} \subseteq A \iff \pi(\{n\}) \subseteq \pi(A) \iff \hat{\pi}(n) \in \pi(A)
\]

Hence \( \pi(A) = \{\hat{\pi}(n) : n \in A\} \).

2. Take any bijection which maps the even integers to some non computable infinite coinfinite set.

3. Let \( M \) be a maximal set. Let \( \pi : \omega \to \omega \) be any bijection such that \( \pi|M = id \). There are continuum many such bijections, one for each permutation of \( \overline{M} \). But for any \( A \in \mathcal{E} \) we have that \( A \cap \overline{M} \) is finite or \( A \cap \overline{M} = \overline{M}^+ \). But this gives us that \( \pi(A) =^* A \). Similarly \( \pi^{-1}(A) =^* A \).

QED

The following theorem shows that the family of hyperhypersimple sets is definable in \( \mathcal{E} \).

**Theorem 101** *(Lachlan)* A is hyperhypersimple iff \( A \) is c.e., \( \overline{A} \) is infinite, and

\[
\mathcal{E} \models \forall B \supseteq A \exists C \supseteq A \ B \cap C = A \text{ and } B \cup C = \omega
\]

**Proof**
Suppose $A$ is not hyperhypersimple and $V$ is a weak array such that $V_e \cap \overline{A} \neq \emptyset$ for all $e$. Define

$$B = A \cup \bigcup_{e \in \omega} (V_e \cap W_e)$$

Suppose for contradiction that $C$ satisfies $B \cap C = A$ and $B \cup C = \omega$. Then for some $e$ we have that $C = W_e$. Let $x \in V_e \cap \overline{A}$. If $x \in W_e$ then $x \in C \cap B$ but this contradicts $B \cap C = A$. If $x \notin W_e$ then $x \notin C$ and $x \notin B$ but this contradicts $B \cup C = \omega$.

Conversely suppose there exists $B$ as above for which there is no $C$. We must show there is a weak array $V$ such that $V_e \cap \overline{A} \neq \emptyset$ for all $e$.

So let $B = \{b_s : s \in \omega\}$ be a 1-1 computable enumeration of $B$ and put $B_s = \{b_t : t < s\}$. Similarly, let $A_s$ be a computable enumeration of $A$.

We will construct $V_{e,s}$ pairwise disjoint subsets of $B$ and meet the requirements:

$$R_e \quad V_e \cap \overline{A} \neq \emptyset$$

We will carry along $g(e, s)$ a gate which we use to let elements into each $V_e$. At stage $s = 0$ as usual we put $V_{e,s} = \emptyset$.

**Stage** $s+1$.

First define

$$g(e, s + 1) = \begin{cases} g(e, s) & \text{if } V_{e,s} \cap \overline{A}_s \neq \emptyset \\ s + 1 & \text{otherwise} \end{cases}$$

Look for the least $e < s$ such that $b_s \leq g(e, s + 1)$ and put $b_s$ into $V_e$, i.e.,

$$V_{e,s+1} = V_{e,s} \cup \{b_s\}$$

**Claim.** $\lim_{s \to \infty} g(e, s) = g(e) < \infty$ and $R_e$ is met.

**Proof**

This is proved by induction on $e$. Choose $s_0$ so that for all $\hat{e} < e$ and $s > s_0$ we have that $g(\hat{e}, s) = g(\hat{e})$ and

$$b_s > \max\{g(\hat{e}) : \hat{e} < e\}$$

Suppose for contradiction that

$$\lim_{s \to \infty} g(e, s) = \infty$$
Define
\[ C = A \cup \bigcup_{s \geq s_0} ([0, g(e, s + 1)] \cap \overline{B}_s) \]
Suppose \( x \in \overline{A} \). Then we claim that
\[ x \in C \text{ iff } x \in \overline{B} \]
This is a contradiction since then \( C \cap B = A \) and \( C \cup B = \omega \).
Suppose \( x \in \overline{B} \). This implies that \( x \in \overline{B}_s \) for all \( s \). But if \( g(e, s) \to \infty \) we have that \( x \in C \).
Suppose \( x \in C \). Then for some \( s \geq s_0 \) we have that \( x \in [0, g(e, s)] \cap \overline{B}_s \) (since we are assuming \( x \notin A \).) If \( x \notin \overline{B} \) then \( x \in B \setminus B_s \). Hence \( x = b_t \) for some \( t \geq s \). But notice that \( b_t = x \leq g(e, s) \leq g(e, t + 1) \). By our choice of \( s_0 \) we have that \( b_t > g(\hat{e}) \) for all \( \hat{e} < e \) and so \( b_t \) will be put into \( V_e \). But \( x = b_t \) was assumed to be an element of \( \overline{A} \). This means that \( g(e, t) \) will never increase again which contradicts it going to \( \infty \).
The reason \( R_e \) is met is because if \( g(e, s) \) stops growing then eventually we stop putting \( b_s \)'s into \( V_e \). Hence \( V_e \) is finite and so it is impossible that \( V_e \subseteq A \).
This proves the Claim and the Theorem.
QED
The following shows that the family of hypersimple sets is not definable in \( \mathcal{E} \).

**Theorem 102** (Martin) There exists a hypersimple set \( A \) and \( \pi \in \text{aut}(\mathcal{E}) \) such that \( \pi(A) \) is not hypersimple.

**Proof**
We will construct the c.e. set \( A \) as usual by constructing a computable increasing sequence \( A_s \). We will construct a computable sequence \( \pi_s \) of bijections of \( \omega \) with the property that \( \pi_s(n) = n \) for every \( n \geq s \). So each \( \pi_s \) is really a finite permutation. \( \pi \) will be the limit of \( \pi_s \).
Let \( W_{e,s}^* \) be defined as follows:
\[ W_{e,s}^* = W_{e,s_0} \text{ where } s_0 \leq s \text{ is the largest } t \leq s \text{ with the property that for distinct } x, y \in W_{e,t} \text{ we have that } D_x \cap D_y = \emptyset. \]
The list \( W_e^* \) automatically contains all strong arrays. Our requirements for this construction include:
\[ R_e \quad W_e^* \text{ infinite } \rightarrow \exists x \in W_e^* \quad F_x \subseteq A \]
The strategy for making sure that $\overline{A}$ is a variant on the Post 2e strategy. At stage $s = 0$ in our construction we have $A_s = \emptyset$ and $\pi_s$ the identity.

**Stage** $s + 1$.

Given $\pi_s$ and $A_s$. We say that $e < s$ requires attention iff

1. $\neg \exists n \in W^*_{e,s} \ D_n \subseteq A_s$
2. $\exists x, y \in W^*_{e,s}$ such that
   a. $x, y \notin A_s$
   b. $\exists n \in W^*_{e,s} \ x \in F_n$
   c. $e < x < y < s, \ e < \pi_s(x), \ e < \pi_s(y)$
   d. i. $e$-state of $x$ at stage $s = e$-state of $y$ at stage $s$
      ii. $e$-state of $\pi_s(x)$ at stage $s = e$-state of $\pi_s(y)$ at stage $s$
   e. $2x < \pi_s(y)$.

The action at this stage is the following. For the least $e < s$ (if any) which requires attention we choose the least $x$ for which there is a $y$ and then we choose the least $y$. For this choice $(e, x, y) = (e_s, x_s, y_s)$ we

a. put $x$ into $A$, $A_{s+1} = A_s \cup \{x_s\}$
   b. put $\pi_{s+1} = \pi_s \circ \text{swap}(x, y)$ where $\text{swap}(x, y)$ refers to the transposition which interchanges $x$ and $y$.

As usual if there is no $e$ which requires attention we do nothing and go onto the next stage.

This ends the construction. Let $Q$ denote the stages $s$ where action takes place at stage $s + 1$. Then

$$A = \{x_s : s \in Q\}$$

We define

$$\pi(u) = \lim_{s \to \infty} \pi_s(u)$$

although at this point we have not proved that this limit always exists. Note the pointwise limit of 1-1 functions must be 1-1 where it is defined.

Note that for $s \in Q$ we have that $\pi_{s+1}(x_s) = \pi_s(y_s)$. Since $x_s$ enters $A$ we have (by 2a) that $x_s$ will never be a $x_t$ or $y_t$ latter. It follows that $\pi(x_s) = \pi_{s+1}(x_s)$. Hence

$$B \overset{df}{=} \{\pi_{s+1}(x_s) : s \in Q\} = \{\pi(x_s) : s \in Q\}$$

73
is well defined and c.e.

**Claim (1)** for any \( n \) we have that \( |B \cap [0, 2n]| \leq n \).

**Proof**
Note that (by 2\( e \)) we have that \( \pi(x_s) = \pi_s(y_s) > 2x_s \). Since each \( x_s \) is distinct the Claim follows.

QED

As we have seen before this implies that \( B \) is not hypersimple. (Proposition 78).

**Claim (2)** \( \lim_{s \to \infty} \pi_s(u) = \pi(u) < \infty \) for every \( u \).

**Proof**
Fix \( s_0 \) so that \( A \cap [0, u] = A_{s_0} \cap [0, u] \). Now the only way that \( \pi_{s+1}(u) \neq \pi_s(u) \) for some \( s > s_0 \) is if \( u = x_s \) or \( u = y_s \). But in either case since \( x_s < y_s \) and \( x_s \) enters \( A \) we have \( A \) changes in the interval \([0, u] \) which is a contradiction.

QED

We don’t know yet that \( \pi \) is onto.

**Claim (3)** For each \( e \)

(a) \( R_e \) is met.

(b) \( \exists s_0 \forall s > s_0 \quad e_s > e \)

**Proof**
This is proved by induction on \( e \).

(a) We may suppose by induction that there exists \( s_0 \) such that \( e_s \geq e \) for all \( s > s_0 \). Suppose \( R_e \) is not met. Then \( W_e^* \) is infinite and for all \( n \in W_e^* \) we have that \( F_n \cap \overline{A} \neq \emptyset \). For each \( n \in W_e^* \) define

\[
x_n = \min(F_n \cap \overline{A})
\]

Since the \( F_n \) are pairwise disjoint all of the \( x_n \) are distinct. Note there exist \( \sigma, \tau \in 2^e \) such that

\[
\exists n \in W_e^* \quad \sigma = \text{final } e\text{-state of } x_n \text{ and } \tau = \text{final } e\text{-state of } \pi(x_n).
\]

Choose \( x_n \) and \( x_m \) such that

1. \( n, m \in W_e^* \)
2. \( e < x_n < x_m \)
3. \( 2x_n < \pi(x_m) \)
4. \( \sigma \) is the final e-state of \( x_n \) and \( x_m \), and

5. \( \tau \) is the final e-state of \( \pi(x_n) \) and \( \pi(x_m) \).

Increase \( s_0 \) (if necessary) so that not only is \( e_s \geq e \) for all \( s \geq s_0 \) but also so that

1. \( n, m \in W_{e,s_0}^* \)
2. \( x_n < x_m < s_0 \) and \( \pi(x_n) < s_0 \) and \( \pi(x_m) < s_0 \)
3. \( \pi_s(x_n) = \pi(x_n) \) and \( \pi_s(x_m) = \pi(x_m) \) all \( s > s_0 \)
4. \( \sigma \) is the e-state of \( x_n \) and \( x_m \) at stage \( s_0 \),
5. \( \tau \) is the e-state of \( \pi(x_n) \) and \( \pi(x_m) \) at stage \( s_0 \) and
6. \( A_{s_0} \cap [0, x_m] = A \cap [0, x_m] \)

Recall that we chose \( x_n, x_m \in \overline{A} \). It is easy to check that \( e \) requires attention at stage \( s_0 \) and \( x_n \) and \( x_m \) witness this fact. But this means that \( x_n \) or some smaller \( x \) enters \( A \). But this contradicts the condition that \( A \) does not change below \( x_m \).

(b) Suppose that \( e_s \geq e \) for all \( s > s_0 \) and \( R_e \) is met. If \( W_e^* \) is infinite, then for some \( x \in W_e^* \) we have that \( F_x \subseteq A \). But this will be seen at some stage and so \( e \) will not require attention after that. If \( W_e^* \) is finite, then suppose that

\[ \cup\{F_x : x \in W_e^*\} \subseteq [0, n]. \]

After we reach a stage \( s > s_0 \) where \( A_s \cap [0, n] = A \cap [0, n] \), then \( e \) will never again require attention because then \( A \) would change beneath \( n \).

QED

Claim (4) \( \pi \) is onto.

Proof
Given \( z \) choose \( s_0 \) so that \( e_s > z \) for all \( s \geq s_0 \). If \( \pi_{s_0}(u) = z \), then \( u \) will never be either \( x_s \) or \( y_s \) for any \( s \geq s_0 \). This is because we required that \( \pi_s(x_s) > e_s > z \) and \( \pi_s(x_s) > e_s > e \). Hence \( \pi(u) = z \).

QED

Claim (5)

(a) \( \forall C \in \mathcal{E} \quad \pi(C) \in \mathcal{E} \)
(b) \( \forall C \in \mathcal{E} \quad \pi^{-1}(C) \in \mathcal{E} \)

Proof

(a) Fix \( s_0 \) so that for all \( s > s_0 \) we have that \( e_s > e \). Then we show that

\[
\pi(W_e) = \bigcup_{s > s_0} \pi_s(W_{e,s})
\]

To see this first suppose \( y \in \pi(W_e) \). Then there exists \( x \in W_e \) with \( \pi(x) = y \) but for all sufficiently large \( s \) we have that \( x \in W_{e,s} \) and \( \pi_s(x) = \pi(x) \) and thus \( y \in \pi_s(W_{e,s}) \).

To see the other inclusion, suppose that \( y \in \pi_s(W_{e,s}) \) for some \( s > s_0 \).
We claim that for every \( t > s \) that \( y \in \pi_t(W_{e,t}) \). This is proved by induction on \( t \).
Suppose that \( \pi_t(u) = y \) for some \( u \in W_{e,t} \). Then \( \pi_{t+1}(u) = \pi_t(u) \) unless \( u = x_t \) or \( u = y_t \) and then \( \pi_{t+1}(x_t) = \pi_t(y_t) \) and \( \pi_{t+1}(y_t) = \pi_t(x_t) \). But since \( x_t \) and \( y_t \) have the same \( e_t \)-type and \( e_t > e \), if one is in \( W_{e,t} \) so is the other.
In either case we have that there exists \( v \in W_{e,t+1} \) with \( \pi_{t+1}(v) = y \). Now to see that \( y \in \pi(W_e) \) suppose that \( \pi(u) = y \) and choose sufficiently large \( t > s_0 \) such that \( \pi_t(u) = \pi(u) = y \). Since \( \pi_t \) is a bijection and \( y \in \pi_t(W_{e,t}) \), it must be that \( u \in W_{e,t} \) and hence \( u \in W_e \).

(b) This is similar, except we use that \( \pi_t(x_t) \) and \( \pi_t(y_t) \) have the same \( e_t \)-type.

QED

Hmwk 30. (Wed 11-24) Prove that there exists a bijection \( \pi : \omega \to \omega \) such that \( \pi(A) \in \mathcal{E} \) for all \( A \in \mathcal{E} \) but \( \pi \notin aut(\mathcal{E}) \). (Hint: use a maximal set.)

Definition 103 For \( A \) and \( B \) predicates over subsets of \( \omega \) or finite products of \( \omega \) we define:

\[
\Pi^0_n = \Sigma^0_n \quad \text{the computable predicates.}
\]

\( A \) is \( \Sigma^0_{n+1} \) iff there exists \( B \) which is \( \Pi^0_n \) and \( A(x) \equiv \exists y \ B(x,y) \).

\( A \) is \( \Pi^0_{n+1} \) iff there exists \( B \) which is \( \Sigma^0_n \) and \( A(x) \equiv \forall y \ B(x,y) \).

\( A \) is \( \Delta^0_n \) iff \( A \) is \( \Sigma^0_n \) and \( A \) is \( \Pi^0_n \).

Note that by DeMorgan’s Laws

\[
\Pi^0_n = \{ \neg A : A \in \Sigma^0_n \}.
\]

\( \Delta^0_n \) is closed under complementation.

76
Proposition 104 Suppose $\Gamma$ is $\Sigma^0_n$, $\Pi^0_n$, or $\Delta^0_n$. Then $\Gamma$ is closed under $\leq_m$, i.e., $A \leq_m B \in \Gamma$ implies $A \in \Gamma$. This implies that if the predicate $B(x,y)$ is in $\Gamma$ and $f$ is a computable function, then $A(x,y) \equiv B(x,f(x))$ is in $\Gamma$. Also, if $A, B \in \Gamma$, then $A \land B$ and $A \lor B$ are both in $\Gamma$. Finally, $\Gamma$ predicates are closed under bounded quantification, e.g., $\exists u < x A(u,x,\ldots)$ and $\forall u < x A(u,x,\ldots)$.

Proposition 105 If $B(x,y)$ in $\Sigma^0_n$, then $A(x) \equiv \exists y B(x,y)$ is in $\Sigma^0_n$. If $B(x,y)$ in $\Pi^0_n$, then $A(x) \equiv \forall y B(x,y)$ is in $\Pi^0_n$.

Proposition 106 $\Sigma^0_n \cup \Pi^0_n \subseteq \Delta_{n+1}^n = \Sigma^0_{n+1} \cap \Pi^0_{n+1}$

Definition 107 We say that $A$ is universal for $\Gamma$ iff
$$\Gamma = \{B : \exists x \ B = A_x\}.$$ We say that $A$ is $m$-complete for $\Gamma$ iff
$$\Gamma = \{B : \ B \leq_m A\}$$

Note that universal for $\Gamma$ implies $m$-complete for $\Gamma$. Also, the complement of a set universal for $\Gamma$ is universal for $\overline{\Gamma}$ and the same for $m$-completeness.

Proposition 108 For each $n > 0$ there is a universal $\Sigma^0_n$ set.

Proposition 109 For each $n > 0$ we have $\text{Red}(\Sigma^0_n)$, $\text{Sep}(\Pi^0_n)$, $\neg\text{Sep}(\Sigma^0_n)$, and $\neg\text{Red}(\Pi^0_n)$.

Proof
See definitions 20. We first show $\text{Red}(\Sigma^0_n)$. Let
$$A(x) \equiv \exists y \ R(x,y) \quad \text{and} \quad B(x) \equiv \exists y \ S(x,y)$$
where $R$ and $S$ are $\Delta^0_n$. Reduce $A$ and $B$ by
$$A_0(x) \equiv \exists y \ (R(x,y) \land \forall z < y \neg S(z,x))$$
and
$$B_0(x) \equiv \exists y \ (S(x,y) \land \forall z \leq y \neg R(z,x))$$
Since $\text{Red}(\Gamma) \rightarrow \text{Sep}(\overline{\Gamma})$ Proposition 22, it follows that $\text{Sep}(\Pi^0_n)$ holds.
To see $\neg Sep(\Sigma^0_n)$, first construct a doubly universal pair $A$ and $B$. These are $\Sigma^0_n$ sets such that for every pair $C$ and $D$ of $\Sigma^0_n$ sets there exists a $u$ with $C = A_u$ and $D = B_u$. To get $A$ and $B$ let $U$ be a universal $\Sigma^0_n$ set. Then define

$$A = \{(\langle x, y \rangle, z) : \langle x, z \rangle \in U\}$$

and

$$B = \{(\langle x, y \rangle, z) : \langle y, z \rangle \in U\}$$

then $u = \langle x, y \rangle$ codes the pair $U_x$ and $U_y$. Now applying reduction to $A$ and $B$ we get $A^0 \subseteq A$ and $B^0 \subseteq B$. Note that this simultaneously reduces all cross sections $A_u$ and $B_u$. Assuming for contradiction that separation holds, let $C$ be $\Delta^0_n$ such that $A^0 \subseteq C$ and $B^0 \subseteq C$. We get a contradiction since, then $C$ would be a universal $\Delta^0_n$ set. This is because if $P$ is $\Delta^0_n$ then there exists $u$ with $A_u = P$ and $B_u = \overline{P}$. But the reduction followed by separation can’t effect the $u$ cross section, so $C_u = P$.

QED

Hmwk 31. (Mon 11-29) Prove there does not exist a universal $\Delta^0_n$ set.

**Lemma 110** $A \subseteq \omega$ is $\Pi^0_2$ iff there exists $P$ computable such that

$$A(x) \equiv \exists^\infty s \ P(s, x)$$

Proof

($\leftarrow$) $\exists^\infty s \ P(s, x)$ iff $\forall t \exists s > t \ P(s, x)$

($\rightarrow$) Suppose

$$A(x) \text{ iff } \forall n \exists m \ R(n, m, x)$$

where $R$ is $\Delta^0_1$. Define $P \subseteq \omega^{<\omega} \times \omega$ by

$$P(\sigma, x) \text{ iff } \forall i < |\sigma| \ [R(i, \sigma(i), x) \text{ and } \forall j < i \ \neg R(i, j, x)]$$

QED

**Theorem 111** (Post) Suppose $A \subseteq \omega$. Then $A$ is $\Delta^0_2$ iff $A \leq_T 0'$

Proof

Suppose $A$ is $\Delta^0_2$. Then by Lemma 110 there exists computable $P(u, x)$ and $Q(v, x)$ such that

$$A(x) \equiv \exists^\infty u \ P(u, x)$$
\[ \neg A(x) \equiv \exists v \ Q(v, x) \]

Now define \( g(x, s) \) as follows. Input \( x, s \) and let \( u_s \) be the maximum \( u \leq s \) such that \( P(u, x) \) (zero if no such \( u \)). Similarly define \( v_s \) to be the maximum \( v \leq s \) such that \( Q(v, x) \). Define

\[
g(x, s) = \begin{cases} 1 & \text{if } u_s \geq v_s \\ 0 & \text{if } u_s < v_s \end{cases}
\]

It is easy to check that

\[ A(x) = \lim_{s \to \infty} g(x, s) \]

and so by the Limit Lemma 62 we have that \( A \leq_T 0' \).

Conversely if \( A \leq_T 0' \) then by the Limit Lemma we have \( g \) computable such that

\[ A(x) = \lim_{s \to \infty} g(x, s) \]

but then

\[ A(x) \equiv \forall^\infty s \ g(x, s) = 1 \equiv \exists^\infty s \ g(x, s) = 1 \]

so \( A \) is \( \Delta^0_2 \).

QED

**Lemma 112**

1. \( A \subseteq \omega \) is \( \Sigma^0_1(B) \) iff \( A \leq_m B' \).
2. \( A \) is \( \Delta^0_2(B) \) iff \( A \leq_T B' \).

**Proof**

\( A \) is \( \Sigma^0_1(B) \) iff there exists a predicate \( R \leq_T B \) such that

\[ A(x) \iff \exists y \ R(x, y) \]

(1) is just a relativization of the standard result that \( 0' \) is \( \Sigma^0_1 \)-m-complete.

(2) is just the relativization of Post’s Theorem 111.

QED

**Theorem 113** *(Post)*

1. \( A \leq_T 0^{(n)} \) iff \( A \) is \( \Delta^0_{n+1} \).
2. \( 0^{(n)} \) is an m-complete \( \Sigma^0_n \)-set.

79
Proof

(1) for $n = 2$: 
\[ A \leq T^0 \iff A \leq T^0' \iff A \leq \Delta^0_2(0') \]
\[ A \text{ is } \Delta^0_2(0') \iff \text{there exists } R_1, R_2 \leq T^0 \text{ such that} \]
\[ A(x) \iff \exists n \forall m R_1(n, m) \]
\[ \neg A(x) \iff \exists n \forall m R_2(n, m) \]

but since $R_1, R_2 \leq T^0' \iff R_1$ and $R_2$ are $\Delta^0_2$, we have that 
\[ A \text{ is } \Delta^0_2(0') \iff A \text{ is } \Delta^0_2 \]

(2) for $n = 2$: 
\[ 0'' \text{ is } \Sigma^0_1(0') \text{ and } m\text{-complete for } \Sigma^0_1(0'). \]
But $\Sigma^0_1(0')$ is $\Sigma^0_2$. This is because 
\[ B \text{ is } \Sigma^0_1(0') \iff \text{there exists } R \leq T^0' \text{ such that} \]
\[ B(x) \iff \exists y R(x, y) \]

But $R \leq T^0' \iff R$ is $\Delta^0_2$. Hence $B$ is $\Sigma^0_2$ iff $B$ is $\Sigma^0_1(0')$.

The proofs for $n > 2$ are analogous.

QED

Hmwk 32. (Wed 12-1) Prove there does not exist $A$ which is $m$-complete for $\Delta^0_2$.

Proposition 114 \( EMP = \{ e : W_e = \emptyset \} \) is $\Pi^0_1$-m-complete.

Proof

\[ e \in EMP \iff \forall x, s \ x \notin W_{e,s} \]

so $EMP$ is $\Pi^0_1$. Let $A$ be $\Pi^0_1$, then there is $R$ computable so that 
\[ A(x) \iff \forall y R(x, y) \]

Using S-n-m Theorem get $f$ computable so that for every $x$
\[ W_{f(x)} = \{ y : \neg R(x, y) \} \]

Then $A(x) \iff f(x) \in EMP$.

QED
Proposition 115 \( \text{TOT} = \text{def} \{ e : W_e = \omega \} \) is \( \Pi^0_2 \)-m-complete.
\( \text{FIN} = \text{def} \{ e : W_e \text{ is finite} \} \) is \( \Sigma^0_2 \)-m-complete.

Proof

\[ e \in \text{TOT} \iff \forall x \exists s x \in W_{e,s} \]
\[ e \in \text{FIN} \iff \exists x \forall y, s (y \in W_{e,s} \rightarrow y < x) \]

so \( \text{TOT} \) is \( \Pi^0_2 \) and \( \text{FIN} \) is \( \Sigma^0_2 \). Now suppose that \( A \) is \( \Pi^0_2 \) we show that

\( (A, \overline{A}) \leq_m (\text{TOT, FIN}) \)

which simultaneously shows that \( \text{TOT} \) is \( \Pi^0_2 \)-m-complete and \( \text{FIN} \) is \( \Sigma^0_2 \)-m-complete. Suppose

\[ A(x) \iff \exists \infty s P(s, x) \]

where \( P \) is \( \Delta^0_1 \). Using S-n-m find a computable function \( f \) so that

\[ W_{f(x)} = \{ t : \exists s > t P(s, x) \} \]

Hence \( A(x) \rightarrow W_{f(x)} = \omega \) while \( \neg A(x) \rightarrow W_{f(x)} \) is finite.

QED

Proposition 116 \( \text{COF} = \text{def} \{ e : \overline{W_e} \text{ is finite} \} \) is \( \Sigma^0_3 \)-m-complete.

Proof

\[ e \in \text{COF} \iff \exists n \forall m > n \exists s m \in W_{e,s} \]

Now suppose that \( A \) is \( \Sigma^0_3 \). Then there exists \( P \) which is \( \Delta^0_1 \) such that

\[ A(x) \iff \exists n \exists \infty m P(n, m, x) \]

Input \( x \) and describe the c.e. set \( B_x \) by using a moving marker construction similar to the construction of a maximal set but simpler. At any stage \( s \) we have that

\[ \overline{B_{x,s}} = \{ p_{0,s} < p_{1,s} < p_{2,s} < \cdots \} \]

We look for the least \( n < s \) (if any) such that \( P(n, s, x) \) and bump the \( n \)th marker, i.e., enumerate \( p_{n,s} \) into \( B_x \), i.e., \( B_{x,s+1} = B_{x,s} \cup \{ p_{n,s} \} \). Note that if \( A(x) \) is true then there exist \( n \) so that the \( n \)th marker is bumped infinitely often and so \( B_x \) is cofinite. On the other hand if \( \neg A(x) \), then each marker eventually stops moving and so \( B_x \) is coinfinite.
By the usual S-n-m argument we can find a computable function \( f \) so that \( B_x = W_{f(x)} \) for all \( x \) and so

\[
A(x) \text{ iff } f(x) \in COF
\]

QED

**Proposition 117** \( REC = \{ e : W_e \text{ is computable } \} \) is \( \Sigma^0_3 \)-m-complete.

Proof

\[
e \in REC \text{ iff } \exists e' \ (W_e \cup W_{e'} = \omega \text{ and } W_e \cap W_{e'} = \emptyset)
\]

and \( W_e \cup W_{e'} = \omega \) is \( \Pi^0_2 \) and \( W_e \cap W_{e'} = \emptyset \) is \( \Pi^0_1 \). To see that it is \( m \)-complete, use a moving marker argument as above. Just add an additional reason to bump the \( e^{th} \) marker to make sure that if \( B_x \) is coinfinite, then for each \( e \)

\[
\psi_e(e) \downarrow \rightarrow \psi_{e,p_e}(e) \downarrow
\]

This guarantees that if \( B_x \) is coinfinite, then \( K \leq_T B_x \).

QED

**Hmwk 33.** (Fri 12-3)

(a) Let \( A \) be an infinite c.e. set. Let

\[
Q_A = \{ e : W_e = A \}
\]

Prove that \( Q_A \) is \( \Pi^0_3 \)-m-complete.

(b) Let \( A \) be a finite nonempty set. Prove that

\[
Q_A = \{ e : W_e = A \}
\]

is \( D(\Sigma^0_1) \)-m-complete, where

\[
D(\Sigma^0_1) = \{ A \cap \overline{B} : A, B \in \Sigma^0_1 \}.
\]

**Lemma 118** Suppose \( A \) is \( \Sigma^0_{k+1} \) then there exists \( B \ \Pi^0_k \) such that

\[
A(x) \text{ iff } \exists y \ B(x,y) \text{ iff } \exists ! y \ B(x,y)
\]
Proof

Suppose

\[ A(x) \iff \exists y \ P(x, y) \]

where \( P \) is \( \Pi^0_k \). Then

\[ A(x) \iff \exists y \ (P(x, y) \land \forall z < y \neg P(x, z)) \]

Define

\[ C(x, y) \iff \forall z < y \neg P(x, z) \]

In case \( k + 1 = 1 \) then \( C \) is \( \Delta^0_1 \). In case \( k + 1 > 1 \) then since \( C \) is \( \Sigma^0_k \) we have by induction a \( \Pi^0_{k-1} \) predicate \( D \) so that

\[ C(x, y) \iff \exists u \ D(x, y, u) \iff \exists! u \ D(x, y, u) \]

Hence

\[ A(x) \iff \exists y \exists u \ (P(x, y) \land D(x, y, u)) \iff \exists! y \exists! u \ (P(x, y) \land D(x, y, u)) \]

so taking \( B(x, (y, u)) \equiv P(x, y) \land D(x, y, u) \) does the trick.

QED

**Proposition 119** (a) \( A \) is \( \Pi^0_3 \) iff there exists \( B \) which is \( \Delta^0_1 \) such that

\[ A(u) \equiv \exists^\infty s \forall n \ B(s, n, u) \]

(b) \( A \) is \( \Pi^0_4 \) iff there exists \( B \) which is \( \Delta^0_1 \) such that

\[ A(x) \equiv \exists^\infty s \exists^\infty t \ B(s, t, x) \]

Proof

(a) Suppose

\[ A(u) \equiv \forall x \exists y \forall z \ R(x, y, z, u) \]

where \( R \) is \( \Delta^0_1 \). Define

\[ Q(x, u) \equiv \exists y \forall z \ R(x, y, z, u) \]

Then by Lemma 118 there is a \( C \) which is \( \Pi^0_1 \) and

\[ Q(x, u) \equiv \exists y \ C(x, y, u) \equiv \exists! y \ C(x, y, u) \]
Hence
\[ A(u) \equiv \forall x \exists! y \ C(x, y, u) \]
\[ A(u) \equiv \exists^\infty \sigma \in \omega^{<\omega} \ \forall i < |\sigma| \ C(i, \sigma(i), u) \]
Note that \( \forall i < |\sigma| \ C(i, \sigma(i), u) \) is \( \Pi_1^0 \) and so there is \( B \) computable so that
\[ \forall n \ B(\sigma, n, u) \equiv \forall i < |\sigma| \ C(i, \sigma(i), u) \]
(b) Suppose
\[ A(u) \equiv \forall x \exists y \ R(x, y, u) \]
where \( R \) is \( \Pi_2^0 \). By Lemma 118 applied to \( \exists y \ R(x, y, u) \) we may assume that
\[ A(u) \equiv \forall x \exists! y \ R(x, y, u) \]
Hence
\[ A(u) \equiv \exists^\infty \sigma \ \forall i < |\sigma| \ R(i, \sigma(i), u) \]
but the predicate
\[ Q(\sigma, u) \equiv \forall i < |\sigma| \ R(i, \sigma(i), u) \]
is \( \Pi_2^0 \) so there exists a computable \( B \) so that
\[ Q(\sigma, u) \equiv \exists^{<\infty} \tau \ B(\sigma, \tau, u) \]
Hence
\[ A(u) \equiv \exists^\infty \sigma \exists^\infty \tau \ B(\sigma, \tau, u) \]
QED

**Hmwk 34.** (Mon 12-6)
Let \( PTIME = \{ e : \psi_e \text{ runs in polynomial time} \} \), i.e., there exists a polynomial \( p(x) \) such that \( \psi_e(x) \) halts in less than \( p(x) \) steps for every \( x \). Prove that \( PTIME \) is \( \Sigma_2^0 \)-m-complete.

**Hmwk 35.** (Wed 12-8) For each \( e \) let \( Q_e = \{ \frac{n}{m+1} : \langle n, m \rangle \in W_e \} \subseteq \mathbb{Q} \).
Define
\[ \Omega = \{ e : Q_e \text{ is order isomorphic to } \omega \} \]
Prove that \( \Omega \) is \( \Pi_3^0 \)-m-complete.

84
Definition 120 \( A \subseteq \omega^\omega \) is \( \Sigma_1 \) iff there exists a computable \( R \subseteq \omega^{<\omega} \times \omega^{<\omega} \) such that
\[
x \in A \equiv \exists y \in \omega^\omega \forall n \in \omega \ R(x|n, y|n).
\]
Similarly \( A \subseteq \omega \) is \( \Sigma_1 \) iff there exists a computable \( R \subseteq \omega \times \omega^{<\omega} \) such that
\[
k \in A \equiv \exists y \in \omega^\omega \forall n \in \omega \ R(k, y|n).
\]
\( \Pi_1 \) sets are the complements of \( \Sigma_1 \) sets and \( \Delta_1 = \Pi_1 \cap \Sigma_1 \).

We can give similar definitions of \( \Sigma_1 \) and \( \Sigma_0 \) and \( \Pi_0 \) for \( \mathcal{X} \) any finite product \( \mathcal{X} = \prod_{i < N} X_i \) where each \( X_i \) is either \( \omega \) or \( \omega^\omega \).

Proposition 121

1. \( \Pi_0 \subseteq \Sigma_1 \)

2. If \( A \subseteq \mathcal{X} \times \omega^\omega \) is \( \Sigma_1 \) then \( B \) is \( \Sigma_1 \) where
\[
B(x) \text{ iff } \exists y \ A(x, y)
\]

3. If \( A \) and \( B \) are \( \Sigma_1 \) then \( A \land B \) and \( A \lor B \) are \( \Sigma_1 \).

4. If \( A \subseteq \omega \times \mathcal{X} \) is \( \Sigma_1 \) then both
\[\begin{align*}
(a) & \quad B(x) \equiv \exists n \in \omega \ A(n, x) \text{ and} \\
(b) & \quad C(x) \equiv \forall n \in \omega \ A(n, x)
\end{align*}\]
are \( \Sigma_1 \).

Proof

(1) trivial

(2) Suppose \( \mathcal{X} = \omega^\omega \) and
\[
A(x, y) \equiv \exists z \ \forall n \ R(x|n, y|n, z|n)
\]
deﬁne
\[
R^*(\sigma, \tau) \text{ iff } R(\sigma, \tau_0, \tau_1) \text{ where } \tau(i) = \langle \tau_0(i), \tau_1(i) \rangle
\]
Then
\[
B(x) \equiv \exists u \ \forall n \ R^*(x|n, u|n)
\]

(3) Suppose
\[
A(x) \equiv \exists y \ C(x, y)
\]
where $C$ and $D$ are $\Pi^0_1$. Then

$$A(x) \lor B(x) \equiv \exists w \ (C(x,w) \lor D(x,w))$$

and

$$A(x) \land B(x) \equiv \exists y \exists z \ (C(x,y) \land D(x,z))$$

(4a) Suppose

$$A(n,x) \equiv \exists y \ \forall m \ R(n,x|m,y|m)$$

Define

$$R^*(x|n,y|m) \text{ iff } R(y(0), x|(n-1), y^*|(n-1)) \text{ where } y^*(i) = y(i+1)$$

Then

$$B(x) \equiv \exists n \ A(n,x) \equiv \exists y \ \forall m \ R^*(x|m,y|m)$$

(4b) Suppose

$$A(n,x) \equiv \exists y \ \forall m \ R(n,x|m,y|m)$$

Define

$$R^*(x|m,z|m) \text{ iff } R(i,x|j,y_i|j) \text{ for each } (i,j) < m \text{ and } y_i(j) = z((i,j)).$$

Then

$$C(x) \equiv \forall n \ A(n,x) \equiv \exists z \ \forall m \ R^*(x|m,z|m)$$

QED

**Proposition 122** Universal $\Sigma^1_1$ sets exists, hence $\Sigma^1_1 \neq \Pi^0_1$.

Proof

Let $U \subseteq \omega \times \mathcal{X} \times \omega^\omega$ be a universal $\Pi^0_1$ set for subsets of $\mathcal{X} \times \omega^\omega$, then

$$V(n,x) \equiv \exists y \ A(n,x,y)$$

is Universal $\Sigma^1_1$.

QED

**Theorem 123** (Tennenbaum) There exists a computable linear order $(\omega, \leq)$ which is isomorphic to $\omega + \omega^*$ with the property that every nonempty c.e. subset of $\omega$ has a $\leq$-least and $\leq$-greatest element.
Proof

Note that $\omega^*$ stands for reverse $\omega$ or equivalently the order type of the negative integers. Let

\[ L = \{ x \in \omega : |\{ y : y < x \}| < \omega \} \quad \text{and} \quad R = \{ x \in \omega : |\{ y : x < y \}| < \omega \} \]

In our construction we make sure that $\omega = L \cup R$ and each is infinite. At stage $s$ we assume that we have (effectively) determined the finite linear order $\preceq |(s \times s)$ and just decide where to put the new element, $s$, of

\[ s + 1 = \{ 0, 1, 2, \ldots, s \} \]

Our requirements are:

\begin{align*}
R_e & \quad \text{We assume at stage $s$ in our construction that some requirements $R_e$, say} \\
W_e & \quad \text{are infinite} \rightarrow W_e \cap L \neq \emptyset \quad \text{and} \quad W_e \cap R \neq \emptyset .
\end{align*}

We assume at stage $s$ in our construction that some requirements $R_e$, say $e \in F_s \subseteq s$, have followers $l_e < s$ and $r_e < s$ which satisfy:

- if $e < e'$ and $e, e' \in F_s$, then $l_e < l_{e'} < r_{e'} < r_e$.

At stage $s + 1$ we look for the smallest $e < s$ (if any) such that

1. $e \notin F_s$ (or equivalently $R_e$ has no followers)

2. there exists $l, r \in W_{e,s}$ such that for every $e' < e$ with $e' \in F_s$ we have that $l_{e'} < l < r < r_{e'}$.

For the smallest such $e$ and smallest such pair $l, r$ we appoint $l = l_e$ and $r = r_e$ the followers of $R_e$ and put

\[ F_{s+1} = \{ e' < e : e' \in F_s \} \cup \{ e \} \]

i.e., we unappoint all followers for $e' > e$. If there is no such $e$ we do not change any followers.

In either case, we put $s$ into the ordering $\preceq |(s \times s)$ in the first gap above all the $l_e$ for $e \in F_{s+1}$ (and therefore, below all the $r_e$ for $e \in F_{s+1}$.)

**Claim.** For each $e$ if $W_e$ is infinite, then $R_e$ obtains permanent followers $l_e$ and $r_e$ and is met.

Proof
Suppose the Claim is true for all $e' < e$. Suppose $s_0$ is a large enough stage so that no $e' < e$ acts after stage $s_0$. Let $e_0$ be the maximum element of $F_{s_0}$ below $e$. Then since $s > s_0$ are put between $l_{e_0}$ and $r_{e_0}$ and $W_e$ is infinite, it must be that some followers are appointed to $R_e$ if it doesn’t already have them. These followers are permanent.

QED

Since infinitely many $W_e$ are infinite and hence acquire permanent followers, it must be that $L$ and $R$ are infinite and therefore the order type we construct is $\omega + \omega^\ast$.

QED

**Corollary 124** (Jockusch) There exists a computable function $f : [\omega]^2 \rightarrow 2$ such that there is no infinite computable $H \in [\omega]^\omega$ such that $f[H]^2$ is constant.

**Proof**

Define

$$f(x, y) = \begin{cases} 1 & \text{if } x < y \rightarrow x < y \\ 0 & \text{if } x < y \rightarrow y < x \end{cases}$$

QED

**Definition 125** $T \subseteq \omega^{<\omega}$ is a well-founded tree iff

(a) $\forall \sigma, \tau \sigma \subseteq \tau \in T \rightarrow \sigma \in T$

(b) $T$ has no infinite branch, i.e., $[T] = \emptyset$ where

$$[T] = \text{def} \{ x \in \omega^\omega : \forall n \ x|n \in T \}.$$ 

**Definition 126** (Kleene-Brouwer ordering) For $\sigma, \tau \in \omega^{<\omega}$

$$\sigma <_{KB} \tau \text{ iff } \sigma \supseteq \tau \text{ or } \exists n < \min(|\sigma|, |\tau|) \ \sigma|n = \tau|n \text{ and } \sigma(n) < \tau(n)$$

$$\sigma \leq_{KB} \tau \text{ iff } \sigma <_{KB} \tau \text{ or } \sigma = \tau$$

**Proposition 127** $\leq_{KB}$ is a computable linear ordering of $\omega^{<\omega}$.

**Theorem 128** (Kleene-Brouwer) Given a tree $T \subseteq \omega^{<\omega}$

$T$ is well-founded iff $(T, \leq_{KB})$ is a well-ordering.
Proof
Suppose that $T$ is not well-founded and $x \in [T]$. Then for each $n$
\[ x|(n+1) <_{KB} x|n \]
and so $(T, \leq_{KB})$ is not a well-ordering.

Conversely, suppose that $(T, \leq_{KB})$ is not a well-ordering and $(\sigma_n \in T : n < \omega)$ is $<_{KB}$-descending, i.e.,
\[ \sigma_{n+1} <_{KB} \sigma_n. \]
Then an easy induction produces $x \in \omega^\omega$ with the property that
\[ \forall n \forall^\infty m \ x|n \subseteq \sigma_m. \]
It follows that $x \in [T]$ and so $T$ is not well-founded.
QED

**Definition 129** For $T \subseteq \omega^{<\omega}$ a tree and $\alpha$ an ordinal we define $T_\alpha \subseteq T$ as follows:
(a) $\sigma \in T_0$ iff $\sigma \in T$ and $\forall n \ \sigma n \not\in T$. (Terminal nodes of $T$.)
(b) $\sigma \in T_\alpha$ iff $\sigma \in T$ and $\forall n \ (\sigma n \in T \rightarrow \sigma n \in T_{<\alpha})$.
(c) $T_{<\alpha} = \text{def} \cup_{\beta < \alpha} T_\beta$.

**Definition 130** For $\sigma \in T$
(a) $\text{rank}_T(\sigma) = \alpha$ where $\alpha$ is the smallest ordinal with $\sigma \in T_\alpha$.
(b) $\text{rank}_T(\sigma) = \infty$ if there is no such $\alpha$.

**Proposition 131** For $T \subseteq \omega^{<\omega}$ a tree, $T$ is well-founded iff $\text{rank}_T(\langle \rangle) < \infty$, i.e., its an ordinal.

Proof
Note that if $\text{rank}_T(\sigma) = \infty$, then there exists $n$ such that $\text{rank}_T(\sigma n) = \infty$. Hence, $\text{rank}_T(\langle \rangle) = \infty$ implies that $T$ has an infinite branch. On the other hand if $\text{rank}_T(\sigma) < \infty$, then for every $n$ with $\sigma n \in T$ we have that
\[ \text{rank}_T(\sigma n) < \text{rank}_T(\sigma) \]
Hence $T$ cannot have an infinite branch.
QED
Definition 132. \( c : T \to \omega \) is a hypcode iff \( T \subseteq \omega^\omega \) is a computable well-founded tree and \( c \) is partial computable map with domain \( T \). Given a hypcode \( c \) we define the sets \( H(c, \sigma) \) as follows by induction on the rank of \( \sigma \). Fix \( U \subseteq \omega \times X \) a universal \( \Sigma^0_1 \) set.

(a) for \( \sigma \in T_0 \) a terminal node of \( T \)
\[
H(c, \sigma) = U_{c(\sigma)}
\]
(b) for \( \sigma \in T \) not terminal and \( c(\sigma) = 0 \)
\[
H(c, \sigma) = \bigcup_{n, \sigma_n \in T} H(c, \sigma_n)
\]
(c) for \( \sigma \in T \) not terminal and \( c(\sigma) > 0 \)
\[
H(c, \sigma) = \bigcap_{n, \sigma_n \in T} H(c, \sigma_n)
\]

\( A \subseteq X \) is hyperarithmetic (HYP) iff there exists a hypcode \( c \) and
\[
A = H(c) =^{ def } H(c, \langle \rangle).
\]

Proposition 133. \( HYP \subseteq \Delta^1_1 \).

Proof

\( x \in H(c) \) iff there exists \( f : T \to \{0, 1\} \) such that

1. \( \forall \sigma \in T_0 \)
\[
f(\sigma) = 1 \text{ iff } x \in U_{c(\sigma)}
\]
2. \( \forall \sigma \in T \setminus T_0 \) if \( c(\sigma) = 0 \) then
\[
f(\sigma) = 1 \text{ iff } \exists n \ (\sigma n \in T \land f(\sigma n) = 1)
\]
3. \( \forall \sigma \in T \setminus T_0 \) if \( c(\sigma) > 0 \) then
\[
f(\sigma) = 1 \text{ iff } \forall n \ (\sigma n \in T \to f(\sigma n) = 1)
\]
4. \( f(\langle \rangle) = 1 \)
It is easy to check that $1 - 4$ are all arithmetic predicates and so $H(c)$ is $\Sigma_1^1$. To see that the complement of $H(c)$ is also $\Sigma_1^1$ just note that $x \notin H(c)$ iff there exists $f : T \to \{0, 1\}$ such that $1, 2, 3,$ and $4'$. $f(\langle \rangle) = 0$.

QED

**Theorem 134 (Kleene-Souslin)**

Suppose $A$ and $B$ are disjoint $\Sigma_1^1$ sets. Then they can be separated by a hyperarithmetic set $C$. Hence $\text{HYP} = \Delta_1^1$.

**Proof**

To simplify the notation we assume that $A, B \subseteq \omega \omega$ although essentially the same proof will work for $A, B \subseteq \omega$ or any $X$. Since $A, B$ are $\Sigma_1^1$ there are computable trees $T^A, T^B \subseteq \bigcup_{n<\omega} \omega^n \times \omega^n$ such that

$x \in A$ iff $\exists y \forall n \ (x|n, y|n) \in T^A$

$x \in B$ iff $\exists z \forall n \ (x|n, z|n) \in T^B$

The fact that $A$ and $B$ are disjoint implies that it is impossible to find $(x, y, z)$ such that $(x|n, y|n) \in T^A$ and $(x|n, z|n) \in T^B$ for all $n$. This tells us how to find our computable well-founded tree $T$.

Given $\rho \in \omega^{<\omega}$ we determine a triple $\text{trip}(\rho) = (\sigma, \tau_1, \tau_2)$ by the rule that $\sigma(i) = \rho(3i)$, $\tau_1(i) = \rho(3i + 1)$, and $\tau_2(i) = \rho(3i + 2)$. We take the natural length functions, namely

- $|\sigma| = |\tau_1| = |\tau_2| = n$ if $|\rho| = 3n$,
- $|\sigma| = n + 1$, $|\tau_1| = |\tau_2| = n$ if $|\rho| = 3n + 1$, and
- $|\sigma| = |\tau_1| = n + 1$, $|\tau_2| = n$ if $|\rho| = 3n + 2$.

Now we define the computable well-founded tree $T \subseteq \omega^{<\omega}$ and hypcode $c : T \to \omega$ as follows:

1. for $\rho \in \omega^{<\omega}$ with length $|\rho| = 3n + 2$ and $\text{trip}(\rho) = (\sigma, \tau_1, \tau_2)$ if
   - (a) $(\sigma|n, \tau_1|n) \in T^A$,
(b) $(\sigma|n, \tau_2) \in T^B$, and
(c) $(\sigma, \tau_1) \not\in T^A$,

then $\rho$ is a terminal node of $T$ and put $c(\rho) = n_0$ where

$$U_{n_0} = \emptyset.$$

2. for $\rho \in \omega^{<\omega}$ with length $|\rho| = 3(n + 1)$ and $\text{trip}(\rho) = (\sigma, \tau_1, \tau_2)$ if
(a) $(\sigma, \tau_1) \in T^A$,
(b) $(\sigma|n, \tau_2|n) \in T^B$, and
(c) $(\sigma, \tau_2) \not\in T^B$,

then $\rho$ is a terminal node of $T$ and put $c(\rho) = n_1$ where

$$U_{n_1} = [\sigma] =_{def} \{x \in \omega^\omega : \sigma \subseteq x\}.$$

3. For any other $\rho$ we put $\rho$ into $T$ iff it is a proper subset of a terminal node of $T$. For these $\rho$ we put $c(\rho) = 0$ if $|\rho| = 3n$ or $|\rho| = 3n + 1$ and put $c(\rho) = 1$ if $|\rho| = 3n + 2$.

Now given $\text{trip}(\rho) = (\sigma, \tau_1, \tau_2)$ define the following sets:

$$A_\rho = \{x \in [\sigma] : \exists y \supseteq \tau_1 \forall n \ (x|n, y|n) \in T^A\}$$

$$B_\rho = \{x \in [\sigma] : \exists z \supseteq \tau_2 \forall n \ (x|n, z|n) \in T^B\}$$

To finish the proof we verify the following:

**Claim.** For each $\rho \in T$ let $\text{trip}(\rho) = (\sigma, \tau_1, \tau_2)$ then

$$A_\rho \subseteq H(c, \rho) \subseteq [\sigma]$$

and

$$B_\rho \subseteq [\sigma] \setminus H(c, \rho)$$

**Proof**

Case $\rho$ a terminal node of $T$.  

92
Note that in case 1 of the definition of $T$, we have that $A_\rho$ is the empty set and $c(\sigma)$ is a code for the empty set and so its OK. In case 2 of the definition of $T$, we have that $B_\rho$ is the empty set and $c(\sigma)$ is a code for $[\sigma]$ and so it's OK.

Case $|\rho| = 3n$ and $\rho$ not terminal.
Note that for nonterminal nodes $\rho$ we have that for every $k$ that $\rho k \in T$. In this case $\text{trip}(\rho k) = (\sigma k, \tau_1, \tau_2)$.

$$A_{\rho k} = [\sigma k] \cap A_\rho$$
$$B_{\rho k} = [\sigma k] \cap B_\rho$$

and by induction

$$A_\rho = \cup_{k<\omega} A_{\rho k} \subseteq \cup_{k<\omega} H(c, \rho k) =^{def} H(c, \rho) \subseteq [\sigma]$$

$(c(\rho) = 0$, so we take unions$)$

$$B_\rho = \cup_{k<\omega} B_{\rho k} \subseteq \cup_{k<\omega} ([\sigma k] \setminus H(c, \rho k)) = [\sigma] \setminus H(c, \rho)$$

The last equality holds because each $H(c, \rho k) \subseteq [\sigma k]$ and $([\sigma k] : k < \omega)$ is a partition of $[\sigma]$.

Case $|\rho| = 3n + 1$ and $\rho$ not terminal.
In this case $\text{trip}(\rho k) = (\sigma, \tau_1 k, \tau_2)$, and also $c(\rho) = 0$, i.e., we take unions.
Note that for every $k$ that $B_{\rho k} = B_\rho$ since neither $\sigma$ nor $\tau_2$ change. Also, by the definition of $A_\rho$ note that

$$A_\rho = \cup_{k<\omega} A_{\rho k}.$$  

Now by inductive hypothesis we have that

$$A_\rho = \cup_{k<\omega} A_{\rho k} \subseteq \cup_{k<\omega} H(c, \rho k) =^{def} H(c, \rho)$$
$$B_\rho \subseteq [\sigma] \setminus H(c, \rho k)$$

for every $k$ so

$$B_\rho \subseteq [\sigma] \setminus H(c, \rho)$$

as was to be proved.
Case $|\rho| = 3n + 2$ and $\rho$ not terminal. In this case $\text{trip}(\rho k) = (\sigma, \tau_1, \tau_2 k)$, and $c(\rho) = 1$, i.e., take intersections. Note that for every $k$ that $A_{\rho k} = A_\rho$ since neither $\sigma$ nor $\tau_1$ change. Now by inductive hypothesis we have that

$$A_\rho \subseteq \cap_{k<\omega} H(c, \rho k) =^{def} H(c, \rho)$$

$$B_\rho = \cup_{k<\omega} B_{\rho k} \subseteq \cup_{k<\omega} [\sigma] \setminus H(c, \rho k) = [\sigma] \setminus H(c, \rho)$$

as was to be proved.

This proves the Claim. However since $A_{} = A$ and $B_{} = B$ the Theorem follows.

QED

94