Sean Williams

Notes 4.3 through 4.15

Prove Real Numbers are uncountable:

N	Real Number:
1	.**<u>5</u>**000...
2	.2**<u>5</u>**00...
3	.32**<u>5</u>**0...
4	.411**<u>6</u>**...

$d_i = n_i(d_i) + 1$ | ith digit of n_i

Using this formula we obtain .6667...-> which is not on the list

---

f: N -> N

	1	2	3	4
F1	**<u>3</u>**	1	2	10
F2	2	**<u>2</u>**	2	2
F3	1	2	**<u>3</u>**	4
g	4	3	4	...

g(1) = F1(1) + 1
g(2) = F2(2) + 1
g(k) = Fk(k) + 1

P(N) = the subsets of the natural numbers

	1	2	3	4
S1	**<u>1</u>**	0	1	1
S2	0	**<u>1</u>**	1	1
S3	1	1	**<u>0</u>**	0
t	0	0	1	...

If k ⊆Sk then k ~Є t
If k ~⊆of Sk then k Є t

Turing Recognizable <=> Enumerable
Turing Recognizable: TM M recognizes the language

try out all strings of length i or less for i steps

ε, 0, 1
ε, 00, 01, 10, 11, 0 1

Whenever a string gets accepted, output it.
(keep track of accepted strings and don't run these again)

If we have an enumerator E, we will create a TM M as follows:
Given an input string s, run the enumerator, of s ever shows up on the output, accept.

<u>Hilbert's Problem:</u>

Polynomial 6x^3*y + 3x^2z - 4xz + 5
This is unsolvable

Equivalent:
Turing's definition of an algorithm: What can be computed on a Turing machine
Church's definition of an algorithm: Using λ calculus

Our intuitive notion of an algorithm <=> Formal notion

<u>Decidable Qs:</u>

Given a graph G, is G connected?

L = { <G> | G is a connected graph }
Where <G> is a string encoding of the graph

A_{DFA} = { <B> | B is an NFA that accepts w}
Simulate w on B, return the end result

A_{NFA} = { <B> | B is an NFA that accepts w}
1) Algorithm to convert NFA => DFA B'
2) Run w on B'

E_{DFA} = { <B> | L(B) = Ø}
k states: all strings of length k or less

EQ_{DFA} = { <A,B> | L(A) = L(B)}
iff L(A) = L(B)
then L(A)   ~L(B) = Ø
~L(A)   L(B) = Ø
if L(A) ≠ L(B) then
can L(A)   ~L(B) = Ø
and ~L(A)   L(B) = Ø
A, create B' to accept ~L(B)