J Bolton Week 2

Languages and Machines

Chap 1: 14, 20, 30, 33, 40, 47

14.

Any equivalence relation will partition the set X into $X_1 ... X_n$ because of Theorem 1.3.4

20.

Suppose all functions are in the countable set $f_1, f_2, ...$
The function

$f(n) = concat(f_n(n), 1)$

cannot be in that set.
OEΔ

30.

$N \rightarrow N$
Basis step:
s(Z) > Z
Inductive step:
if a > b, then s(a) > b and s(a) > s(b)
C

[NxN] → [NxN]
Basis:
[s(Z), x] > [Z, y]
[Z, s(Z)] > [Z, Z]
Inductive step:
if [a,b] > [c,d], then:

[a, s(b)] > [c, s(d)]
[a, s(b)] > [c, d]
[s(a), b] > [c, d]
[s(a), b] > [s(c), d]
[s(a), s(b)] > [s(c), s(d)]
if a != c then: [a, b] > [c, s(d)]

C

33.

Basis: If n = 1, then m * n = m
Recursive step: m * s(n) = m + (m * n)
C

40.

Prove $1 + 2^n < 3^n$ for all n > 2

Basis: $1 + 2^3 = 9 < 3^3 = 27$
Inductive:

$1+2^{n+1} < 3^{n+1}$
$1 + 2 * 2^n < 3 * 3^n$
$1 + 2^n + 2^n < 3^n + 3^n + 3^n$

$1 + 2^n < 3^n$ {inductive hypothesis}
$2^n < 3^n + 3^n$

thus,
$1+2^{n+1} < 3^{n+1}$

OEΔ

47.
Basis: 2(1) – 1 = 1
Induction:
2(n+1) – 1
2n + 2 – 1
2n+1

For the binary tree to be strict, we must add two new nodes in each step – in other words, n+1 will be equivalent to n+2, since the node we are adding onto must either have two children or be a leaf.

Thus, if we assume 2n-1 to be true, we can add two nodes for (n+1) to get 2n-1 + 2

2n-1+2 =
2n+1

OEΔ