CS 331 HW 8
Cahleh Brancheau

68/100

57)

Proof:

Let TM M_A, M_B st $L(M_A) = A$, $L(M_B) = B$

Machine M

```
1. Input x
2. Simulate M_A on x
3. If M_A rejects REJect
4. Simulate M_B on x
5. I M_B rejects REJECT
```

Then what?

16/20

58-60)

Prove $A \Rightarrow B \Rightarrow C \Rightarrow A$

58)

$A \Rightarrow B$

Assume A, Let TM M_A

TM M_B

```
1. input X
2. Run M_A on X
3. IF M_A ~~accepts output X~~
```

20/20

A = dom f

What if M_A rejects?

59)

$B \Rightarrow C$

Assume B.

TM M_C

```
1. input X
2. run M_B on f(X)
3. If M_B accepts output f(x)
```

A = range f

60)

$C \Rightarrow A$

Assume C.

TM M_A

```
1. input X
```

6/20

I Don't how

61)

1)

Assume A is decidable. Let TM M_A that decides A.

Case 1: finite

Trivial

Case 2: infinite

```
TM $M_F$
1. input n
2. Run $M_A$ on each elt in assending order in the std enumeration of $\{0,1\}^*$
   counting the accepts
3. output n'th accepted string
```

2)

Assume M_F.

Case 1: finite

Trivial

Case 2: infinite

```
TM $M_{A2}$
1. input x
2. run $M_F$ on each $n \in \mathbb{N}$
3. If output matches x ACCEPT
4. If output comes after x in the std enumeration of $\{0,1\}^*$
   REJECT
```

□

62)

I Don't Know

6/20

63)

ICA $A = (Q', \delta')$

head & current state

not head

- - . .	0, x	1, x	1, q_5	0, x	1, x	- - -

$Q' = \Gamma \times (Q \cup \{x\})$

$\forall\ p', q', r' \in Q'$

$\delta'(p', q', r') = \begin{cases} \text{- If } q' \text{ contains } x \text{ in right hand of tuple, do nothing} \\ \text{- If } q' \text{ contains state compute } \delta(\text{state, value}) \text{ in } q' \end{cases}$

write (new value, x), change x in right or Left cell to the state obtained from δ(state, value).

64)

ICA A

- - -	q^0	q^0	q^0	q^1	q^2	q^3	q^0	q^0	q^0	. - -

TM M

. - -	⊔	⊔	⊔	$(q_1, -, -, -)$	$(q^2, -, -, -)$	$(q_3, -, -, -)$	⊔	⊔	⊔	- - -
	-3	-2	-1	0	1	2	3	4	5	
				↑						

All q_0's become ⊔, and all cells become 4-tuples. (current state, Left state, right state, result state). head starts at 0. The head will move left, read the current state move right and write the state in the Left state position. It will write ⊔ if the Left state was ⊔. Repeat this in the other direction and then compute the new state based off of the three states. finaly write the new state in the result state Position. scan right repeating this process until a blank space is found. Scan Left replacing the current state with the result state and the Left, right, and result state positions with dashes in each cell until a blank is found. One cycle has now been completed. Repeat process for each time step. the start position is always the Left most non-blank Cell.

20/20