

## 24.118: Paradox and Infinity

Ryan Lacey <rlacey@mit.edu>

Collaborator(s): Evan Thomas

April 17, 2014

1. (a) 2 states

```
0 1 1 r 0
0 _ 1 l 1
1 1 1 l 1
1 _ * r halt
```

- (b) 3 states

```
0 1 2 r 0
0 _ _ l 1
1 2 1 r 2
1 1 1 l 1
1 _ * r halt
2 1 1 r 2
2 _ 1 l 1
```

- (c) Write the sequence of  $k$  1s.

```
1 _ 1 r 2
2 _ 1 r 3
3 _ 1 r 4
.
.
.
k-1 _ 1 r k
k _ 1 l k-1
```

Traverse backwards to beginning of list.

```
k-1 1 1 1 k-2
k-2 1 1 1 k-3
.
.
.
1 1 1 * halt
```

Stop upon return to first state. Requires  $k$  states.

(d)  $M^{BB}$  requires  $k + b + c + d$  states

$k$  states to write  $k$  ones

$b=3$  states to duplicate the initial sequence

$c$  states for Busy Beaver

$d = 1$  state to write an additional 1

2. The blanks would be used as word stops. In other words the sequence `_1_` would represent the first symbol while `_11_` would represent the second symbol. Combinations of these sequences could be used with the same computational power as the language with two symbols.
3. The Busy Beaver function would use the halting Oracle to test a sequence before execution. If the sequence will halt, then we run the sequence and can track the productivity, otherwise it moves on to the next sequence. Doing this over all of the sequences and taking the max productivity found would satisfy the Busy Beaver function.