Decrementing Numbers

- Now, let's build a TM that decrements a number.
- We'll assume that
  - the tape head points at the start of a number,
  - there is at least one blank on each side of the number.
- The tape head will end at the start of the number after decrementing it.
- If the number is 0, then the subroutine should somehow signal this rather than making the number negative.
Decrementing Numbers

- Now, let's build a TM that decrements a number.
- We'll assume that
  - the tape head points at the start of a number,
  - there is at least one blank on each side of the number.
- The tape head will end at the start of the number after decrementing it.
- If the number is 0, then the subroutine should somehow signal this rather than making the number negative.
Decrementing Numbers

• Now, let's build a TM that decrements a number.
• We'll assume that
  – the tape head points at the start of a number,
  – there is at least one blank on each side of the number.
• The tape head will end at the start of the number after decrementing it.
• If the number is 0, then the subroutine should somehow signal this rather than making the number negative.
Decrementing Numbers

- Now, let's build a TM that decrements a number.
- We'll assume that
  - the tape head points at the start of a number,
  - there is at least one blank on each side of the number.
- The tape head will end at the start of the number after decrementing it.
- If the number is 0, then the subroutine should somehow signal this rather than making the number negative.
Decrementing Numbers

• Now, let's build a TM that decrements a number.
• We'll assume that
  – the tape head points at the start of a number,
  – there is at least one blank on each side of the number.
• The tape head will end at the start of the number after decrementing it.
• If the number is 0, then the subroutine should somehow signal this rather than making the number negative.
Decrementing Numbers

• Now, let's build a TM that decrements a number.
• We'll assume that
  – the tape head points at the start of a number,
  – there is at least one blank on each side of the number.
• The tape head will end at the start of the number after decrementing it.
• If the number is 0, then the subroutine should somehow signal this rather than making the number negative.
Decrementing Numbers

• Now, let's build a TM that decrements a number.
• We'll assume that
  - the tape head points at the start of a number,
  - there is at least one blank on each side of the number.
• The tape head will end at the start of the number after decrementing it.
• If the number is 0, then the subroutine should somehow signal this rather than making the number negative.
Decrementing Numbers

• Now, let's build a TM that decrements a number.
• We'll assume that
  – the tape head points at the start of a number,
  – there is at least one blank on each side of the number.
• The tape head will end at the start of the number after decrementing it.
• If the number is 0, then the subroutine should somehow signal this rather than making the number negative.

1 0 1
Decrementing Numbers

• Now, let's build a TM that decrements a number.
• We'll assume that
  – the tape head points at the start of a number,
  – there is at least one blank on each side of the number.
• The tape head will end at the start of the number after decrementing it.
• If the number is 0, then the subroutine should somehow signal this rather than making the number negative.
Decrementing Numbers

• Now, let's build a TM that decrements a number.

• We'll assume that
  – the tape head points at the start of a number,
  – there is at least one blank on each side of the number.

• The tape head will end at the start of the number after decrementing it.

• If the number is 0, then the subroutine should somehow signal this rather than making the number negative.
Decrementing Numbers

• Now, let's build a TM that decrements a number.
• We'll assume that
  - the tape head points at the start of a number,
  - there is at least one blank on each side of the number.
• The tape head will end at the start of the number after decrementing it.
• If the number is 0, then the subroutine should somehow signal this rather than making the number negative.
Decrementing Numbers

• Now, let's build a TM that decrements a number.

• We'll assume that
  - the tape head points at the start of a number,
  - there is at least one blank on each side of the number.

• The tape head will end at the start of the number after decrementing it.

• If the number is 0, then the subroutine should somehow signal this rather than making the number negative.
Decrementing Numbers

• Now, let's build a TM that decrements a number.
• We'll assume that
  − the tape head points at the start of a number,
  − there is at least one blank on each side of the number.
• The tape head will end at the start of the number after decrementing it.
• If the number is 0, then the subroutine should somehow signal this rather than making the number negative.
Decrementing Numbers

- Now, let's build a TM that decrements a number.
- We'll assume that
  - the tape head points at the start of a number,
  - there is at least one blank on each side of the number.
- The tape head will end at the start of the number after decrementing it.
- If the number is 0, then the subroutine should somehow signal this rather than making the number negative.

![Diagram of a tape with a number 100]
Decrementing Numbers

- Now, let's build a TM that decrements a number.
- We'll assume that
  - the tape head points at the start of a number,
  - there is at least one blank on each side of the number.
- The tape head will end at the start of the number after decrementing it.
- If the number is 0, then the subroutine should somehow signal this rather than making the number negative.
Decrementing Numbers

- Now, let's build a TM that decrements a number.
- We'll assume that
  - the tape head points at the start of a number,
  - there is at least one blank on each side of the number.
- The tape head will end at the start of the number after decrementing it.
- If the number is 0, then the subroutine should somehow signal this rather than making the number negative.

1 0 0
Decrementing Numbers

- Now, let's build a TM that decrements a number.
- We'll assume that
  - the tape head points at the start of a number,
  - there is at least one blank on each side of the number.
- The tape head will end at the start of the number after decrementing it.
- If the number is 0, then the subroutine should somehow signal this rather than making the number negative.
Decrementing Numbers

- Now, let's build a TM that decrements a number.
- We'll assume that
  - the tape head points at the start of a number,
  - there is at least one blank on each side of the number.
- The tape head will end at the start of the number after decrementing it.
- If the number is 0, then the subroutine should somehow signal this rather than making the number negative.
Decrementing Numbers

• Now, let's build a TM that decrements a number.
• We'll assume that
  – the tape head points at the start of a number,
  – there is at least one blank on each side of the number.
• The tape head will end at the start of the number after decrementing it.
• If the number is 0, then the subroutine should somehow signal this rather than making the number negative.
Decrementing Numbers

• Now, let's build a TM that decrements a number.
• We'll assume that
  – the tape head points at the start of a number,
  – there is at least one blank on each side of the number.
• The tape head will end at the start of the number after decrementing it.
• If the number is 0, then the subroutine should somehow signal this rather than making the number negative.
Decrementing Numbers

• Now, let's build a TM that decrements a number.
• We'll assume that
  – the tape head points at the start of a number,
  – there is at least one blank on each side of the number.
• The tape head will end at the start of the number after decrementing it.
• If the number is 0, then the subroutine should somehow signal this rather than making the number negative.

1 0 0
Decrementing Numbers

• Now, let's build a TM that decrements a number.
• We'll assume that
  – the tape head points at the start of a number,
  – there is at least one blank on each side of the number.
• The tape head will end at the start of the number after decrementing it.
• If the number is 0, then the subroutine should somehow signal this rather than making the number negative.
Decrementing Numbers

• Now, let's build a TM that decrements a number.
• We'll assume that
  – the tape head points at the start of a number,
  – there is at least one blank on each side of the number.
• The tape head will end at the start of the number after decrementing it.
• If the number is 0, then the subroutine should somehow signal this rather than making the number negative.
Decrementing Numbers

• Now, let's build a TM that decrements a number.
• We'll assume that
  – the tape head points at the start of a number,
  – there is at least one blank on each side of the number.
• The tape head will end at the start of the number after decrementing it.
• If the number is 0, then the subroutine should somehow signal this rather than making the number negative.
Decrementing Numbers

• Now, let's build a TM that decrements a number.
• We'll assume that
  – the tape head points at the start of a number,
  – there is at least one blank on each side of the number.
• The tape head will end at the start of the number after decrementing it.
• If the number is 0, then the subroutine should somehow signal this rather than making the number negative.
Decrementing Numbers

go to the end of the number;
if (every digit was 0) {
    signal that we're done;
}
while (the current digit is 0) {
    set the current digit to 9;
    back up one digit;
}
decrement the current digit;
go to the start of the number;
<p>| ... | 0 | 2 | 0 | 0 | ... |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th>0</th>
<th>2</th>
<th>0</th>
<th>0</th>
<th></th>
<th></th>
</tr>
</thead>
</table>

... start

...
Non-zero?

0 → 0, R
Non-zero?

\[ 0 \rightarrow 0, R \]
\[ 1 \rightarrow 1, R \]
\[ 2 \rightarrow 2, R \]
\[ \ldots \]
\[ 9 \rightarrow 9, R \]

Start
To End

0 → 0, R
1 → 1, R
2 → 2, R
...
9 → 9, R

Non-zero?
Non-zero? 0 → 0, R 1 → 1, R 2 → 2, R ... 9 → 9, R

To End

... 0 2 0 0 ...
0 → 0, R
1 → 1, R
... 
9 → 9, R

Non-zero?

To
End

start

0 → 0, R

... 0 2 0 0 ...
0 → 0, R
1 → 1, R
...
9 → 9, R

To End

Non-zero?

0 → 0, R
1 → 1, R
2 → 2, R
...
9 → 9, R

start

... 0 2 0 0 ...
To End

Non-zero?

0 → 0, R
1 → 1, R
...
9 → 9, R

start

0 → 0, R

0 2 0 0
To
End
Non-
zero?

0 → 0, R
1 → 1, R
... 
9 → 9, R

0 → 0, R
1 → 1, R
2 → 2, R
... 
9 → 9, R

□ → □, L

... 0 2 0 0 ...

start
To End

Non-zero?

Wrap Zeros

0 → 0, R
1 → 1, R
...
9 → 9, R

1 → 1, R
2 → 2, R
...
9 → 9, R

start

0 → 0, R

0 → 9, L

... 0 2 9 9 ...

...
To End

0 → 0, R
1 → 1, R
...
9 → 9, R

Non-zero?

1 → 1, R
2 → 2, R
...
9 → 9, R

start

Wrap Zeros

0 → 9, L
1 → 0, L
2 → 1, L
3 → 2, L
...
9 → 8, L

... 0 2 9 9 ...

...
To End

Non-zero?

Wrap Zeros

0 → 0, R
1 → 1, R
...
9 → 9, R

1 → 1, R
2 → 2, R
...
9 → 9, R

0 → 0, R

0 → 9, L
1 → 0, L
2 → 1, L
3 → 2, L
...
9 → 8, L

... 0 1 9 9 ...

...
0 → 0, R
1 → 1, R
... 
9 → 9, R

To End

□ → □, L

Wrap Zeros
0 → 9, L

1 → 0, L
2 → 1, L
3 → 2, L
... 
9 → 8, L

Non-zero?

start

Back Home

0 → 0, R
1 → 1, R
2 → 2, R
... 
9 → 9, R

... 0 1 9 9 ...

...
... 0 1 9 9 ...
To End

0 → 0, R
1 → 1, R
...
9 → 9, R

Non-zero?

1 → 1, R
2 → 2, R
...
9 → 9, R

start

0 → 0, R

Wrap Zeros

0 → 9, L

1 → 0, L
2 → 1, L
3 → 2, L
...
9 → 8, L

done!

□ → □, R

Back Home

0 → 0, L
1 → 1, L
...
9 → 9, L

... 0 1 9 9 ...
Non-zero?

To End

0 → 0, R
1 → 1, R
...
9 → 9, R

start
Non-zero?

0 → 0, R
1 → 1, R
2 → 2, R
...
9 → 9, R

Wrap Zeros

0 → 9, L
1 → 0, L
2 → 1, L
3 → 2, L
...
9 → 8, L

Back Home

0 → 0, L
1 → 1, L
...
9 → 9, L

done!

0 → 9, L
1 → 0, L
2 → 1, L
3 → 2, L
...
9 → 8, L

Back Home
To End

Wrap Zeros

Non-zero?

start

done!

Back Home
Non-zero?

To End

Wrap Zeros

Back Home

done!

start

Non-zero?

0 → 0, R
1 → 1, R

... 9 → 9, R

0 → 0, R

1 → 1, R
2 → 2, R

... 9 → 9, R

□ → □, L

0 → 9, L

1 → 0, L
2 → 1, L
3 → 2, L

... 9 → 8, L

0 → 0, L
1 → 1, L

... 9 → 9, L

◻ ◻

done!
To End

Non-zero?

Wrap Zeros

Back Home

0 → 0, R
1 → 1, R
...
9 → 9, R

0 → 0, R
1 → 1, R
2 → 2, R
...
9 → 9, R

1 → 0, L
2 → 1, L
3 → 2, L
...
9 → 8, L

0 → 0, L
1 → 1, L
...
9 → 9, L

done!

… 0 1 9 8 …
0 → 0, R
1 → 1, R
...
9 → 9, R

0 → 0, R
1 → 1, R
2 → 2, R
...
9 → 9, R

Start

Non-zero?

To End

□ → □, L

Wrap Zeros

0 → 9, L

1 → 0, L
2 → 1, L
3 → 2, L
...
9 → 8, L

done!

Back Home

0 → 0, L
1 → 1, L
...
9 → 9, L

... 0 1 9 8 ...

...
Non-zero?

0 → 0, R
1 → 1, R
2 → 2, R
... 
9 → 9, R

To End

Wrap Zeros

0 → 9, L
1 → 0, L
2 → 1, L
3 → 2, L
... 
9 → 8, L

done!

Back Home

0 → 0, L
1 → 1, L
... 
9 → 9, L

... 0 1 9 8 ...
0 → 0, R
1 → 1, R

Non-zero?

To End

□ → □, L

Wrap Zeros

0 → 9, L
1 → 0, L
2 → 1, L
3 → 2, L
... 9 → 8, L
... 9 → 9, R

done!

Back Home

0 → 0, L
1 → 1, L
... 9 → 9, L

start
0 → 0, R
1 → 1, R
... 9 → 9, R

start

0 → 0, R

Non-zero?

To End

□ → □, L

Wrap Zeros

0 → 9, L

1 → 0, L
2 → 1, L
... 9 → 8, L

0 → 0, L
1 → 1, L
... 9 → 9, L

Back Home

done!

□ → □, R

... 0 0 0 0 0 ...
0 → 0, R
1 → 1, R
...
9 → 9, R

Non-zero?

To End

0 → 0, R
1 → 1, R
2 → 2, R
...
9 → 9, R

start

0 → 0, R

Wrap Zeros

0 → 9, L
1 → 0, L
2 → 1, L
3 → 2, L
...
9 → 8, L

done!

Back Home

0 → 0, L
1 → 1, L
...
9 → 9, L

◻ ◻

[newline]

... 0 0 0 0
Non-zero?

To End

Wrap Zeros

Back Home

done!

Start

Non-zero?

→ 0, R
1 → 1, R
... 
9 → 9, R

1 → 1, R
2 → 2, R
... 
9 → 9, R

→ 0, R

→ 0 → 9, L
1 → 0, L
2 → 1, L
3 → 2, L
... 
9 → 8, L

→ 0 → 0, L
1 → 1, L
... 
9 → 9, L

→ 0 → 0, R
1 → 1, R
... 
9 → 9, R

→ 0 → 0, R
1 → 1, R
... 
9 → 9, R

→ 0 → 0, R
1 → 1, R
... 
9 → 9, R

→ 0 → 0, R
1 → 1, R
... 
9 → 9, R

→ 0 → 0, R
1 → 1, R
... 
9 → 9, R

→ 0 → 0, R
1 → 1, R
... 
9 → 9, R

→ 0 → 0, R
1 → 1, R
... 
9 → 9, R

→ 0 → 0, R
1 → 1, R
... 
9 → 9, R

→ 0 → 0, R
1 → 1, R
... 
9 → 9, R

→ 0 → 0, R
1 → 1, R
... 
9 → 9, R
Non-zero?

To End

Wrap Zeros

Back Home

0 → 0, R
1 → 1, R
...
9 → 9, R

1 → 1, R
2 → 2, R
...
9 → 9, R

start

Non-zero?

0 → 0, R

square → square, L

done!

square → square, R

square → square, L

0 → 0, L
1 → 0, L
2 → 1, L
3 → 2, L
...
9 → 8, L

1 → 1, L
...
9 → 9, L

0 → 0, L
1 → 1, L
...
9 → 9, L

... 0 0 0 0 ...

...
... 0 0 0 0 0 ...

- **Non-zero?**
  - 0 → 0, R
  - 1 → 1, R
  - 2 → 2, R
  - 3 → 3, R
  - 4 → 4, R
  - 5 → 5, R
  - 6 → 6, R
  - 7 → 7, R
  - 8 → 8, R
  - 9 → 9, R
- **To End**
  - □ → □, L
- **Wrap Zeros**
  - 0 → 9, L
  - 1 → 0, L
  - 2 → 1, L
  - 3 → 2, L
  - 4 → 3, L
  - 5 → 4, L
  - 6 → 5, L
  - 7 → 6, L
  - 8 → 7, L
  - 9 → 8, L
- **Back Home**
  - 0 → 0, R
  - □ → □, L
  - 0 → 0, L
- **done!**
  - □ → □, R
  - 0 → 0, L
  - 1 → 1, L
  - 2 → 2, L
  - 3 → 3, L
  - 4 → 4, L
  - 5 → 5, L
  - 6 → 6, L
  - 7 → 7, L
  - 8 → 8, L
  - 9 → 9, L
Non-zero?

To End

0 → 0, R
1 → 1, R
...
9 → 9, R

0 → 0, R
1 → 1, R
2 → 2, R
...
9 → 9, R

Non-zero?

0 → 0, R

done!

Wrap Zeros

0 → 9, L
1 → 0, L
2 → 1, L
3 → 2, L
...
9 → 8, L

0 → 0, L
1 → 1, L
...
9 → 9, L

Back Home

0 → 0, L

Back Home

0 → 0, R

... 0 0 0 0 0 ...

...
The diagram illustrates a state machine with the following transitions:

- **Non-zero?**:
  - 0 → 0, R
  - 1 → 1, R
  - 2 → 2, R
  - ... → ...
  - 9 → 9, R

- **To End**:
  - □ → □, L
  - 0 → 0, R
  - 1 → 1, R
  - 2 → 2, R
  - ... → ...
  - 9 → 9, R

- **Wrap Zeros**:
  - 0 → 9, L
  - 1 → 0, L
  - 2 → 1, L
  - 3 → 2, L
  - ... → ...
  - 9 → 8, L

- **Back Home**:
  - 0 → 0, R
  - 1 → 1, R
  - 2 → 2, R
  - ... → ...
  - 9 → 9, R

- **done!**:
  - □ → □, R

The diagram starts at the non-zero state and transitions to the to end state if the digit is non-zero. If the digit is 0, it transitions to wrap zeros, and from there, it can transition back to the back home state or continue to wrap zeros. The state of wrap zeros loops back to itself, and the back home state loops back to itself as well.
Non-zero?

0 → 0, R
1 → 1, R
... 9 → 9, R

To End

□ → □, L

Wrap Zeros

0 → 9, L
1 → 0, L
2 → 1, L
3 → 2, L
... 9 → 8, L

n = 0

[start]

□ → □, R

Back Home

0 → 0, R

0 → 0, L

Non-zero?

□ → □, L

Back Home

□ → □, R

done!

0 → 0, L
1 → 1, L
2 → 2, L
3 → 3, L
... 9 → 9, L

... 0 0 0 0 0 ...

...
TM Subroutines

• Sometimes, a subroutine needs to report back some information about what happened.

• Just as a function can return multiple different values, we'll allow subroutines to have different “done” states.

• Each state can then be wired to a different state, so a TM using the subroutine can control what happens next.