Another TM Design

- We've designed a TM for \( \{0^n1^n \mid n \in \mathbb{N} \} \).
- Consider this language over \( \Sigma = \{0, 1\} \):
  \[
  L = \{ \ w \in \Sigma^* \mid w \text{ has the same number of 0s and 1s } \}
  \]
- This language is also not regular, but it is context-free.
- How might we design a TM for it?
A Caveat

... 0 0 0 1 1 1 1 1 0 ...
A Caveat

... 0 0 1 1 1 1 1 0 ...

...
A Caveat

... 0 0 1 1 1 1 1 0 ...

...
A Caveat

... 0 0 1 1 1 1 1 0 ...

...
A Caveat
A Caveat
A Caveat

\[ \ldots \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0 \quad \ldots \]
A Caveat
A Caveat
A Caveat
A Caveat

How do we know that this blank isn't one of the infinitely many blanks after our input string?
A Caveat
A Caveat

... 0 1 1 0 ...

...
A Caveat
A Caveat
A Caveat
A Caveat
A Caveat

How do we know that this blank isn't one of the infinitely many blanks after our input string?
A Caveat
A Caveat

How do we know that this blank isn't one of the infinitely many blanks after our input string?
One Solution
One Solution

... \times 0 0 1 1 1 1 1 0 ...

...
One Solution

\[ \cdots \times 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ \cdots \]
One Solution

| ... | $\times$ | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | ... |

...
One Solution

... × 0 0 × 1 1 1 0 ...
One Solution

... × 0 0 × 1 1 1 1 0 ...
One Solution

... × 0 0 × 1 1 1 1 0 ...
One Solution

... \times 0 0 \times 1 1 1 0 ...
One Solution

... × 0 0 × 1 1 1 0 ...

...
One Solution

... × 0 0 × 1 1 1 0 ...
One Solution
One Solution

... × × 0 × 1 1 1 1 0 ...

...
One Solution

... × × 0 × 1 1 1 1 0 ...
One Solution
One Solution

... × × 0 × × 1 1 0 ...
One Solution

... × × 0 × × 1 1 0 ...
One Solution
One Solution

... × × 0 × × 1 1 0 ...

...
One Solution

... × × 0 × × 1 1 0 ...

...
One Solution

... × × 0 × × 1 1 0 ...
One Solution

| ... | × | × | 0 | × | × | 1 | 1 | 0 | ... |
One Solution
One Solution

| ... | ×××××××110 | ... |
One Solution

... × × × × × × 1 1 0 ...

...
One Solution

... × × × × × × × 1 0 ...

...
One Solution

... × × × × × × × 1 0 ...

...
Find 0/1
Find 0/1

0 → x, R

... 0 0 1 1 1 1 1 1 0 0 ...
Find 0/1

0 → x, R

Find 1

0 → 0, R

... \times 0 1 1 1 1 1 1 0 0 ...

start
Find 0/1

0 → ×, R

Find 1

0 → 0, R

... × 0 1 1 1 1 1 0 0 ...
Find 0/1

0 → x, R

Find 1

1 → x, L

0 → 0, R

0 1 1 1 1 1 0 0
Find 0/1

0 → x, R

Find 1

1 → x, L

0 → 0, R

Go home

0 → 0, L
1 → 1, L
x → x, L

0 1 1 1 1 0 0 ...
Find 0/1

0 → x, R

Find 1

1 → x, L

0 → 0, R

Go home

0 → 0, L
1 → 1, L
x → x, L

... x 0 x 1 1 1 1 0 0 ...

...
Find 0/1

[Diagram]

Go home

... $\times 0 \times 1 1 1 1 0 0 ...$
\[
\begin{array}{cccccc}
0 & 0 & 1 & 1 & 1 & 1 \\
\times & 0 & \times & 1 & 1 & 0 \; 0
\end{array}
\]
Start

Find 0/1

0 → ×, R

Find 1

0 → ×, R
1 → ×, L

Go home

0 → 0, L
1 → 1, L
× → ×, L

... × × × × 1 1 1 1 0 0 ...
Find 0/1

0 \rightarrow \times, R

\times \rightarrow \times, R

Find 1

0 \rightarrow \times, R
0 \rightarrow 0, R
\times \rightarrow \times, R

1 \rightarrow \times, L

Go home

0 \rightarrow 0, L
1 \rightarrow 1, L
\times \rightarrow \times, L

\ldots \times \times \times 1 1 1 1 0 0 \ldots
Find 0/1

0 → ×, R

Find 1

0 → ×, R
1 → ×, L

Go home

0 → 0, L
1 → 1, L
× → ×, L

start

… × × × × × 1 1 0 0 …
Find 0/1

Start

Find 1

Go home

... x x x x x 1 1 0 0 ...

0 → 0, L
1 → 1, L
x → x, L

0 → x, R
1 → x, L

0 → 0, R
x → x, R

□ → □, R
\begin{itemize}
\item \textbf{Find 0/1:} $x \rightarrow x, R$
\item Go home: $0 \rightarrow 0, L$
\item \textbf{Find 1:} $0 \rightarrow x, R$
\item \textbf{Go home:} $1 \rightarrow 1, L$
\item \textbf{Find 1:} $x \rightarrow x, R$
\item \textbf{Find 1:} $1 \rightarrow x, L$
\end{itemize}
Find 0/1

0 → x, R

Find 1

1 → x, L

Go home

0 → 0, L
1 → 1, L
x → x, L
Find 0/1

Find 1

Go home

\[ \begin{array}{c}
\times \rightarrow \times, R \\
0 \rightarrow \times, R \\
1 \rightarrow \times, L \\
\square \rightarrow \square, R \\
0 \rightarrow 0, L \\
1 \rightarrow 1, L \\
\times \rightarrow \times, L \\
\end{array} \]
Find 0/1

Find 1

Go home

\[\begin{align*}
0 & \rightarrow x, R \\
1 & \rightarrow x, R \\
\square & \rightarrow \square, R \\
\end{align*}\]

\[\begin{align*}
0 & \rightarrow 0, L \\
1 & \rightarrow 1, L \\
x & \rightarrow x, L \\
\end{align*}\]
0 → ×, R
× → ×, R

Find 0

1 → 1, R
× → ×, R

Find 1

0 → ×, R
1 → ×, L

Go home

0 → 0, L
1 → 1, L
× → ×, L

start

Find 0/1

x → x, R
The diagram represents a finite automaton with states. The start state is labeled "0/1" and transitions based on the input symbols 0 and 1. The automaton moves through states labeled "Find 0", "Find 1", and ends in the state "Go home".

The transitions are as follows:
- From 0/1: 
  - 0 → 0, R
  - 1 → ×, R
  - × → ×, R
- From Find 0: 
  - 0 → ×, R
  - 1 → 1, R
  - × → ×, R
- From Find 1: 
  - 0 → ×, L
  - 1 → 1, L
  - × → ×, L
- From Go home: 
  - 0 → 0, L
  - 1 → 1, L
  - × → ×, L
Find 0

0 → 0, L
1 → 1, L
× → ×, L

Find 1

0 → ×, R
1 → ×, L

Find 0/1

× → ×, R

Go home

0 → ×, L
1 → ×, L

start

... x x x x x x 1 x x ...
\[ \begin{array}{c}
\text{Find 0} \\
1 \rightarrow 1, R \\
\times \rightarrow \times, R \\
\hline
\text{Find 0/1} \\
\times \rightarrow \times, R \\
\hline
\text{Find 1} \\
1 \rightarrow \times, L \\
0 \rightarrow \times, R \\
\hline
\text{Find 0} \\
\times \rightarrow \times, R \\
\hline
\text{Go home} \\
0 \rightarrow 0, L \\
1 \rightarrow 1, L \\
\times \rightarrow \times, L \\
\hline
\end{array} \]
Go home

Find 0

1 → 1, R
x → x, R

Find 0

0 → x, L

Find 1

1 → x, L

Find 1

0 → 0, R
x → x, R

Find 0

0 → 0, R
1 → 1, R
x → x, L

Go home

0 → 0, L
1 → 1, L
x → x, L

start

x → x, R

x → x, R

... × × × × × × × × 1 × × 0 ...

...
\[
\begin{array}{c}
\text{Find 0} \\
0 \rightarrow 0, \text{L} \\
1 \rightarrow 1, \text{L} \\
\times \rightarrow \times, \text{L}
\end{array}
\]

\[
\begin{array}{c}
\text{Find 1} \\
0 \rightarrow 0, \text{R} \\
1 \rightarrow 1, \text{R} \\
\times \rightarrow \times, \text{R}
\end{array}
\]

\[
\begin{array}{c}
\text{Go home} \\
0 \rightarrow 0, \text{L} \\
1 \rightarrow 1, \text{L} \\
\times \rightarrow \times, \text{L}
\end{array}
\]

\[
\begin{array}{c}
\text{Find 0/1} \\
\times \rightarrow \times, \text{R}
\end{array}
\]

\[
\begin{array}{c}
\text{start} \\
\times \rightarrow \times, \text{R}
\end{array}
\]
Remember that all missing transitions implicitly reject.
Constant Storage

- Sometimes, a TM needs to remember some additional information that can't be put on the tape.
- In this case, you can use similar techniques from DFAs and introduce extra states into the TM's finite-state control.
- The finite-state control can only remember one of finitely many things, but that might be all that you need!