Designing to a Specification: Example 1.4. Again.

- Ignore Leading 0s
- Require at least one 1
- After a 1, you may not have a single trailing 0.

Idea 1: A string of the form (Zero or more 0s) (then a final 1) minimally satisfies all conditions. Call that the "basic condition." Status.

Start

\[ M = \begin{array}{c}
0 \\
1 \\
\end{array} \]

\[ L(M) = \{ x \in \{0, 1\}^* : M \text{ accept } x \} \]

Also, accept anything that meets these specifications:

- an odd # of trailing 0s
- \( A' \)

Design for the original spec, excluding only strings that end in 10.

\[ W = 01001111 \]

When the last char is a 1, we don't care about the previous character.

\[ X = 011000 \in L(M) \]

\[ Y = 0110 \notin L(M) \]

\[ Z = 011000 \in L(M) \]

Last char read was a 1.

If a property (a, b, or c) is violated, then that means a slipshy x that the machine accepts but it shouldn't.

This M is sound for the spec A'

\[ L(M) \subseteq A \]

but it is not "comprehensive".

Last 2 chars read were --
Spec B: \( \Sigma = \{0, 1, 2\} \):
- \(X \) must have length \( \geq 3\)
- The 3rd from last char must be a 1.

State label yikes last 3 chars read

---

Idea:

Since initial Qs don't matter, "000" can be the start state

A smaller "Machine":

---

Motivation read §1.2 and 1.3 of [this page](#).