The Guide to the Lava Diagram
Hi everybody!
As you probably noticed on Problem Set Nine – and on the practice final exams – we love asking questions about “The Lava Diagram.”
The Lava Diagram is this Venn diagram showing the relationships between the regular, decidable, and recognizable languages.
(In case you're wondering, this isn't really called “The Lava Diagram.” That's just a fun name some students came up with a while back. I liked it, so I've kept using it ever since!)
Usually, we'll ask a question of the form “take this group of languages and place each one of them into the diagram in the proper place.”
This question is designed to test your intuition for what the different classes of languages mean. The first time you see a problem like this, it can be tricky!
However, there are a bunch of useful intuitions that can help guide you while working on these problems. We'll go and talk about them by working through these four languages here.

\[ L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \]
\[ L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \]
\[ L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \]
\[ L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \} \]
Let's start by looking at this language $L_1$ and seeing where it should go.

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$

$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$

$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
There are a couple of different strategies you can use to work through these problems, but the one we find the most useful is to start from the outside and work inward.

\[
L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \\
L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \\
L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \\
L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}
\]
That is, we're going to start off with $L_1$ in the **ALL** section, then try to see how far down we can push it into the Lava Diagram.

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |L(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |L(M)| = 2 \}$

$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$

$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
The very first question we should ask ourselves, therefore, is whether this language belongs to \textbf{RE}.

\begin{align*}
L_1 &= \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \\
L_2 &= \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \\
L_3 &= \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \\
L_4 &= \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}
\end{align*}
So what exactly is the class \textbf{RE}?

\begin{itemize}
  \item $L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$
  \item $L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$
  \item $L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$
  \item $L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
\end{itemize}
When we first defined \textbf{RE}, we said that it was the class of all the recognizable languages.

\begin{align*}
L_1 & = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \\
L_2 & = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \\
L_3 & = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \\
L_4 & = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}
\end{align*}
This means that we could try to think about **RE** as “the class of problems with recognizers.”

\[
L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \\
L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \\
L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \\
L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}
\]
However, later on, we saw a different definition of \textbf{RE}, which I think is actually a lot more useful here.

\[
\begin{align*}
L_1 &= \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \\
L_2 &= \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \\
L_3 &= \{ \text{a}^n\text{b}^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \\
L_4 &= \{ \text{a}^n\text{b}^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}
\end{align*}
\]
Specifically, we saw that \textit{RE} is the class of languages that have verifiers.

- \( L_1 \): \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} 
- \( L_2 \): \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} 
- \( L_3 \): \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} 
- \( L_4 \): \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}
If you think back to what a verifier for a language is supposed to do, at a high level, it's really an "answer checker."

\[ \begin{align*}
L_1 &= \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \\
L_2 &= \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \\
L_3 &= \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \\
L_4 &= \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}
\end{align*} \]
Specifically, a verifier is supposed to take in a string and a certificate, then see whether the certificate proves whether the string is in the language.

\[ L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |L(M)| \geq 2 \} \]
\[ L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |L(M)| = 2 \} \]
\[ L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \]
\[ L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \} \]
In that sense, you can think of the \textbf{RE} languages this way: they're the languages where, for any string in the language, there's some way to \textit{prove} that the string is indeed in the language.

\begin{itemize}
  \item $L_1$ (?)
  \item $L_2 = \{ \langle M \rangle \mid M$ is a TM and $|\mathcal{L}(M)| = 2 \}$
  \item $L_3 = \{ a^n b^n \mid n \in \mathbb{N}$ and $n > 1000 \}$
  \item $L_4 = \{ a^n b^n \mid n \in \mathbb{N}$ and $n \leq 1000 \}$
\end{itemize}
Turns out, this provides an amazingly good intuition for the RE languages. A language is in RE if and only if, whenever you have a string in the language, there's some way to prove it's in the language.

\[ L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \]
\[ L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \]
\[ L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \]
\[ L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \} \]
We're going to use this intuition a ton when working through these problems. It's definitely worth making a note of this technique!

\[ L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \]
\[ L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \]
\[ L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \]
\[ L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \} \]
So let's go focus our attention to the particular language $L_1$ we have right now.

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$

$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$

$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
Imagine you have a string in $L_1$. What does that string look like?

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$

$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$

$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
Well, according to the definition of the language, any string in $L_1$ must encode a TM where $|\mathcal{L}(M)| \geq 2$.

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$

$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$

$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
So what exactly does that mean?

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$
$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$
$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$
$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
Well, the language of a TM is the set of strings that it accepts.

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$

$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$

$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
So, if $|\mathcal{L}(M)| \geq 2$, it means that $M$ accepts at least two strings.

$L_1 = \{ \langle M \rangle | M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle | M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$

$L_3 = \{ a^n b^n | n \in \mathbb{N} \text{ and } n > 1000 \}$

$L_4 = \{ a^n b^n | n \in \mathbb{N} \text{ and } n \leq 1000 \}$

RE: Languages with Verifiers

Given any string $w \in L$, could you prove that $w \in L$?
Given any string $w \in L$, could you prove that $w \in L$?

So we can think of $L_1$ as “the language of TMs that accept at least two strings.”

$L_1 = \{ \langle M \rangle | M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$
$L_2 = \{ \langle M \rangle | M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$
$L_3 = \{ a^n b^n | n \in \mathbb{N} \text{ and } n > 1000 \}$
$L_4 = \{ a^n b^n | n \in \mathbb{N} \text{ and } n \leq 1000 \}$
With that in mind, let's think about whether this language is in \textit{RE} or not.

\begin{itemize}
    \item $L_1$ (?)
\end{itemize}

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$

$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$

$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
Let's imagine that we have a random TM and we are convinced that it accepts at least two strings.

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$
$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$
$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$
$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
Is there something we could do to prove that it accepts at least two strings?

\[
\begin{align*}
L_1 &= \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \\
L_2 &= \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \\
L_3 &= \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \\
L_4 &= \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}
\end{align*}
\]
In other words, if we came across someone who was skeptical that the machine actually accepts at least two strings, could we convince them that the machine indeed does accept at least two strings?

\[
L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |L(M)| \geq 2 \}
\]
\[
L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |L(M)| = 2 \}
\]
\[
L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}
\]
\[
L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}
\]
In this case, the answer is yes!

\[ L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \]
\[ L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \]
\[ L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \]
\[ L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \} \]
If we happened to know at least two strings that the machine accepted, we could just run the machine on both those strings and watch it accept them.

$\mathcal{L}_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$

$\mathcal{L}_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$

$\mathcal{L}_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$

$\mathcal{L}_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
Anyone who was initially skeptical that our TM accepted at least two strings would definitely be convinced at that point. They just watched the TM accept at least two strings!

\[ L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \]
\[ L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \]
\[ L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \]
\[ L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \} \]
So, going off this intuition, we can be reasonably confident that the language $L_1$ is indeed in $\text{RE}$.

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$

$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$

$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
At this point we haven't ruled out the possibility that it's also in \( R \) or is regular, but it's almost certainly not outside \( \text{RE} \).

\[
L_1 = \{ \langle M \rangle | M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}
\]
\[
L_2 = \{ \langle M \rangle | M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}
\]
\[
L_3 = \{ a^n b^n | n \in \mathbb{N} \text{ and } n > 1000 \}
\]
\[
L_4 = \{ a^n b^n | n \in \mathbb{N} \text{ and } n \leq 1000 \}
\]
Although the question here was just to go and place \( L_1 \), it's not a bad idea to think about how we'd actually go and build a verifier for \( L_1 \).

\[
L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \\
L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \\
L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \\
L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}
\]
The idea would go something like this.

\[ L_1 = \{ \langle M \rangle | M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \]
\[ L_2 = \{ \langle M \rangle | M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \]
\[ L_3 = \{ a^n b^n | n \in \mathbb{N} \text{ and } n > 1000 \} \]
\[ L_4 = \{ a^n b^n | n \in \mathbb{N} \text{ and } n \leq 1000 \} \]
We can prove that our TM $M$ accepts at least two strings by telling our verifier what two strings $M$ is going to accept.

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$
$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$
$L_3 = \{ ab^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$
$L_4 = \{ ab^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
To ensure that our verifier doesn't go into an infinite loop (remember – verifiers aren't allowed to loop!), we can also give the verifier the number of steps it's going to take for $M$ to accept.

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$
$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$
$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$
$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
Given any string $w \in L$, could you prove that $w \in L$?

So the verifier would take in as input the TM $M$, two strings $w_1$ and $w_2$, and a number of steps $n$, and could run $M$ on the strings $w_1$ and $w_2$ for up to $n$ steps.

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |L(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |L(M)| = 2 \}$

$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$

$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
If M accepts both $w_1$ and $w_2$ within that many steps, then the verifier is convinced that M definitely accepts at least two strings.

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$

$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$

$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
If that doesn't happen, the verifier isn't sure of what the answer is. Maybe \( M \) does accept two strings and we gave the verifier the wrong strings, or maybe we gave it the wrong number of steps.

\[
L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \\
L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \\
L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \\
L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}
\]
If you wanted to write this up as a formal proof, it's a good exercise! For now, though, we're just going to continue working through figuring out where this language goes on the Lava Diagram.

$L_1 = \{ \langle M \rangle | M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle | M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$

$L_3 = \{ a^n b^n | n \in \mathbb{N} \text{ and } n > 1000 \}$

$L_4 = \{ a^n b^n | n \in \mathbb{N} \text{ and } n \leq 1000 \}$
Okay! So at this point we know that $L_1$ is in $\text{RE}$.

$L_1 = \{ \langle M \rangle | M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle | M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$

$L_3 = \{ a^n b^n | n \in \mathbb{N} \text{ and } n > 1000 \}$

$L_4 = \{ a^n b^n | n \in \mathbb{N} \text{ and } n \leq 1000 \}$

**RE: Languages with Verifiers**

Given any string $w \in L$, could you prove that $w \in L$?
The next step is to determine whether it's also in class $\mathbf{R}$. 

$L_1 = \{ \langle M \rangle | M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} 

L_2 = \{ \langle M \rangle | M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} 

L_3 = \{ a^n b^n | n \in \mathbb{N} \text{ and } n > 1000 \} 

L_4 = \{ a^n b^n | n \in \mathbb{N} \text{ and } n \leq 1000 \}
So what exactly is the class $R$?

$L_1 = \{ \langle M \rangle | M \text{ is a TM and } |L(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle | M \text{ is a TM and } |L(M)| = 2 \}$

$L_3 = \{ a^n b^n | n \in \mathbb{N} \text{ and } n > 1000 \}$

$L_4 = \{ a^n b^n | n \in \mathbb{N} \text{ and } n \leq 1000 \}$
Well, we defined it to be the class of all decidable languages.

\[
\begin{align*}
L_1 &= \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \\
L_2 &= \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \\
L_3 &= \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \\
L_4 &= \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}
\end{align*}
\]
That means that it's the class of all languages that have deciders.

\[ L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \]
\[ L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \]
\[ L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \]
\[ L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \} \]
You can reason about whether a language belongs to class $\mathbf{R}$ by thinking about whether you could build a decider for it.

### Languages

- **$L_1$**
  \[ L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |L(M)| \geq 2 \} \]

- **$L_2$**
  \[ L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |L(M)| = 2 \} \]

- **$L_3$**
  \[ L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \]

- **$L_4$**
  \[ L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \} \]
There's an alternative perspective that I think is a bit easier to use, though.

\[ L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \]

\[ L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \]

\[ L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \]

\[ L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \} \]
On Problem Set Nine, there’s a problem entitled “Double Verification.”

\[ L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \]
\[ L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \]
\[ L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \]
\[ L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \} \]
We asked you to prove this statement:
If \( L \in \text{RE} \) and \( \overline{L} \in \text{RE} \), then \( L \in \text{R} \).
What exactly does that mean?

\[
L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |L(M)| \geq 2 \} \\
L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |L(M)| = 2 \} \\
L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \\
L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}
\]
If $L \in \text{RE}$ and $\overline{L} \in \text{RE}$, then $L \in \text{R}$.

From what we've talked about so far, you probably have a slightly better feel for what it means for $L$ to be in $\text{RE}$.

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$

$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$

$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
If $L \in \mathcal{RE}$ and $L \in \mathcal{RE}$, then $L \in \mathcal{R}$.

But what exactly does it mean for the complement of $L$ to be in $\mathcal{RE}$?

$L_1 = \{ \langle M \rangle | M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$
$L_2 = \{ \langle M \rangle | M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$
$L_3 = \{ a^n b^n | n \in \mathbb{N} \text{ and } n > 1000 \}$
$L_4 = \{ a^n b^n | n \in \mathbb{N} \text{ and } n \leq 1000 \}$
If $L \in \text{RE}$ and $\overline{L} \in \text{RE}$, then $L \in \text{R}$.

Going off of our proof-based intuition, if the complement of $L$ is in \text{RE}, it means that given any string $w$ that is \textit{not} in $L$, there's a way to prove it's not in $L$.

\[
L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}
\]
\[
L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}
\]
\[
L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}
\]
\[
L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}
\]
This turns out to be a great way of intuiting the class $\mathbf{R}$. A language belongs to $\mathbf{R}$ if it's in $\mathbf{RE}$, and for any string that isn't in the language, there's a way to prove it's not in the language.

$\mathbf{R}$: Languages with Deciders

In addition to the $\mathbf{RE}$ requirements, given any string $w \notin L$, could you prove that $w \notin L$?

$\mathbf{RE}$: Languages with Verifiers

Given any string $w \in L$, could you prove that $w \in L$?

$L_1 = \{ \langle M \rangle \mid M$ is a TM and $|\mathcal{L}(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle \mid M$ is a TM and $|\mathcal{L}(M)| = 2 \}$

$L_3 = \{ a^n b^n \mid n \in \mathbb{N}$ and $n > 1000 \}$

$L_4 = \{ a^n b^n \mid n \in \mathbb{N}$ and $n \leq 1000 \}$
(Although we only had you prove the forward direction of the implication in the Double Verification problem, turns out the reverse direction holds as well. This gives an exact characterization of $\mathbf{R}$!

**RE: Languages with Verifiers**

Given any string $w \in L$, could you prove that $w \in L$?

**R: Languages with Deciders**

In addition to the RE requirements, given any string $w \notin L$, could you prove that $w \notin L$?

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$

$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$

$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
Now, let's jump back to our particular language $L_1$ here and use this intuition to think about whether or not it belongs to class $R$.

**RE: Languages with Verifiers**
Given any string $w \in L$, could you prove that $w \in L$?

**R: Languages with Deciders**
In addition to the RE requirements, given any string $w \notin L$, could you prove that $w \notin L$?

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$

$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$

$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
**RE: Languages with Verifiers**
Given any string \( w \in L \), could you prove that \( w \in L \)?

**R: Languages with Deciders**
In addition to the RE requirements, given any string \( w \notin L \), could you prove that \( w \notin L \)?

---

\[ L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \]
\[ L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \]
\[ L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \]
\[ L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \} \]
In other words, imagine you have TM where $|\mathcal{L}(M)| < 2$.

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$

$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$

$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
That means that $M$ must accept either no strings at all or just one string.

(Do you see why?)

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$

$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$

$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
So now the question is the following: if you have a TM that accepts either no strings or just one string, could you prove it to someone who was skeptical but honest?
This is going to be a bit tricky.

\[ L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \]

\[ L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \]

\[ L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \]

\[ L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \} \]
REG

**RE: Languages with Verifiers**
Given any string \( w \in L \), could you prove that \( w \in L \)?

**R: Languages with Deciders**
In addition to the **RE** requirements, given any string \( w \notin L \), could you prove that \( w \notin L \)?

\[
\begin{align*}
L_1 &= \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \\
L_2 &= \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \\
L_3 &= \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \\
L_4 &= \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}
\end{align*}
\]
As we've seen before, though, we know that the only general way to find out what a TM will do on a string is to run the TM on that string and see what happens.

\[
L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}
\]

\[
L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}
\]

\[
L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}
\]

\[
L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}
\]
So if we want to convince someone that a TM doesn't accept infinitely many different strings, we're out of luck! In the general case, we'd have to run the TM on all those strings...

**RE: Languages with Verifiers**

Given any string $w \in L$, could you prove that $w \in L$?

---

**R: Languages with Deciders**

In addition to the RE requirements, given any string $w \notin L$, could you prove that $w \notin L$?

---

$L_1 = \{ \langle M \rangle \mid M$ is a TM and $|\mathcal{L}(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle \mid M$ is a TM and $|\mathcal{L}(M)| = 2 \}$

$L_3 = \{ a^n b^n \mid n \in \mathbb{N}$ and $n > 1000 \}$

$L_4 = \{ a^n b^n \mid n \in \mathbb{N}$ and $n \leq 1000 \}$
RE: Languages with Verifiers

Given any string $w \in L$, could you prove that $w \in L$?

R: Languages with Deciders

In addition to the RE requirements, given any string $w \notin L$, could you prove that $w \notin L$?

... and given that there are infinitely many of them, we'll never finish checking them all.

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$
$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$
$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$
$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
Based on the intuition that a language is in $\mathbf{R}$ if we can always prove it when strings aren't in the language, we'd suspect that this language is not in $\mathbf{R}$.

**RE: Languages with Verifiers**
Given any string $w \in L$, could you prove that $w \in L$?

**R: Languages with Deciders**
In addition to the RE requirements, given any string $w \notin L$, could you prove that $w \notin L$?

So, based on the intuition that a language is in $\mathbf{R}$ if we can always prove it when strings aren't in the language, we'd suspect that this language is not in $\mathbf{R}$.

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$

$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$

$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
To actually go and prove this, we could use some kind of self-reference trick and build a machine that asks whether it's going to accept at least two strings, then does the opposite.

\[
L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}
\]
\[
L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}
\]
\[
L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}
\]
\[
L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}
\]
In fact, that's such a good exercise that you should stop reading this and go do it right now. The Guide to Self-Reference might help you there.

\[
L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \\
L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \\
L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \\
L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}
\]
So did you go prove that yet? If not, you really should think about doing so. It's a great exercise!

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$

$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$

$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
So at this point we've got this language settled in the right place. It's in \( \text{RE} \), but it's not in \( \text{R} \).

\[
L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}
\]
\[
L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}
\]
\[
L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}
\]
\[
L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}
\]
Before we move on to the next language, I wanted to take a minute to address a common question we get on problems like these.

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |L(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |L(M)| = 2 \}$

$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$

$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
If you look at the description of the language, you can see that it says something about TMs that accept at least two strings.

\[ L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |L(M)| \geq 2 \} \]
\[ L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |L(M)| = 2 \} \]
\[ L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \]
\[ L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \} \]
A lot of people ask – “Isn't it really easy to build a TM that accepts at least two strings? So shouldn't this be decidable? Or even regular?”

**REG**

**RE: Languages with Verifiers**
Given any string $w \in L$, could you prove that $w \in L$?

**ALL**

$L_1 = \{ \langle M \rangle | M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle | M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$

$L_3 = \{ a^nb^n | n \in \mathbb{N} \text{ and } n > 1000 \}$

$L_4 = \{ a^nb^n | n \in \mathbb{N} \text{ and } n \leq 1000 \}$

**R: Languages with Deciders**
In addition to the **RE** requirements, given any string $w \notin L$, could you prove that $w \notin L$?

A lot of people ask – “Isn't it really easy to build a TM that accepts at least two strings? So shouldn't this be decidable? Or even regular?”
The answer to that question is “yes, and no.”

**RE: Languages with Verifiers**

Given any string \( w \in L \), could you prove that \( w \in L \)?

**R: Languages with Deciders**

In addition to the RE requirements, given any string \( w \notin L \), could you prove that \( w \notin L \)?

\[
L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}
\]
\[
L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}
\]
\[
L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}
\]
\[
L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}
\]
It is indeed possible to build a TM that accepts at least two strings.

\[ L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \]
\[ L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \]
\[ L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \]
\[ L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \} \]
We can do that by just building a TM that accepts everything, for example.

**R: Languages with Deciders**

In addition to the RE requirements, given any string \( w \notin L \), could you prove that \( w \notin L \)?

**RE: Languages with Verifiers**

Given any string \( w \in L \), could you prove that \( w \in L \)?

\[
\begin{align*}
L_1 &= \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \\
L_2 &= \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \\
L_3 &= \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \\
L_4 &= \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}
\end{align*}
\]
But notice that this problem isn't asking whether you can build this machine. It's a question about the language of all TMs with this particular property.

\[ L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \]
\[ L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \]
\[ L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \]
\[ L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \} \]
In that sense, the question is really asking "how hard is it to tell whether a random TM actually does accept at least two strings?"

**L**
\[ L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \]
\[ L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \]
\[ L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \]
\[ L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \} \]
That question – the question of checking whether a TM has some behavior – is typically much, much harder than the problem of building a TM with that behavior.

**REG**

**RE**: Languages with Verifiers

Given any string \( w \in L \), could you prove that \( w \in L \)?

**R**: Languages with Deciders

In addition to the **RE** requirements, given any string \( w \notin L \), could you prove that \( w \notin L \)?

**ALL**

\[
\begin{align*}
L_1 &= \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \\
L_2 &= \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \\
L_3 &= \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \\
L_4 &= \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}
\end{align*}
\]
RE: Languages with Verifiers
Given any string $w \in L$, could you prove that $w \in L$?

R: Languages with Deciders
In addition to the RE requirements, given any string $w \notin L$, could you prove that $w \notin L$?

Keep that in mind going forward – the question is to determine whether an arbitrary string is in the language, not to try to find a string that happens to be in the language.

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$
$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$
$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$
$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
And with that said, let's move on to the second language!

\[ L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \]

\[ L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \]

\[ L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \]

\[ L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \} \]
Before I talk about this particular problem, take a few minutes to think about where you believe this should go in the Lava Diagram. Once you've done that, let's rejoin and keep talking.

$\text{ALL}$

$\text{RE}$: Languages with Verifiers

Given any string $w \in L$, could you prove that $w \in L$?

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$

$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$

$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$

$\text{REG}$

$\text{R}$

$\text{RE}$
Did you actually go and think about it? If not, you should. Like, seriously. It's good practice.

**RE: Languages with Verifiers**

Given any string $w \in L$, could you prove that $w \in L$?

**R: Languages with Deciders**

In addition to the RE requirements, given any string $w \notin L$, could you prove that $w \notin L$?

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$

$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$

$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
Okay! So now you've given it your best shot. Let's see where this one goes.

**L**: Languages with Deciders

In addition to the **RE** requirements, given any string $w \notin L$, could you **prove** that $w \notin L$?

**RE**: Languages with Verifiers

Given any string $w \in L$, could you **prove** that $w \in L$?

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$

$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$

$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
As before, we're going to start on the outside and move inward. Initially, we won't make any assumptions about where this particular language goes.

**RE: Languages with Verifiers**

Given any string \( w \in L \), could you prove that \( w \in L \)?
Our first question is to determine whether this language belongs to RE or not.

**RE: Languages with Verifiers**

Given any string \( w \in L \), could you prove that \( w \in L \)?

**R: Languages with Deciders**

In addition to the RE requirements, given any string \( w \notin L \), could you prove that \( w \notin L \)?

\[
\begin{align*}
L_1 &= \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \\
L_2 &= \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \\
L_3 &= \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \\
L_4 &= \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}
\end{align*}
\]
To do so, we're going to ask whether, given a random string in the language, it's possible to prove it's in the language.

**RE**: Languages with Verifiers
Given any string $w \in L$, could you prove that $w \in L$?

**R**: Languages with Deciders
In addition to the RE requirements, given any string $w \notin L$, could you prove that $w \notin L$?

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$
$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$
$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$
$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
Looking over the definition of the language, we see that this is the language of all TMs whose language has size two.

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \quad R: \text{Languages with Deciders}

In addition to the RE requirements, given any string $w \notin L$, could you prove that $w \notin L$?

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \quad \ast L_2$ (？)

$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \quad \ast L_1$

$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \} \quad \text{RE: Languages with Verifiers}

Given any string $w \in L$, could you prove that $w \in L$?
This means that this is the language of all TMs that accept exactly two strings.

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$

$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$

$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
So now we ask – if had a TM and you knew for a fact that it accepted exactly two strings, could you prove it?

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$

$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$

$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
This turns out to be a lot harder than just checking if a TM accepts at least two strings.

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$

$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$

$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
To show that a TM accepts exactly two strings, we need to show that it accepts at least two strings (that's something we can prove), but also that it doesn't accept anything else.

- \( L_1 = \{ \langle M \rangle | M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \)
- \( L_2 = \{ \langle M \rangle | M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \)
- \( L_3 = \{ a^n b^n | n \in \mathbb{N} \text{ and } n > 1000 \} \)
- \( L_4 = \{ a^n b^n | n \in \mathbb{N} \text{ and } n \leq 1000 \} \)
The problem is that to show that a TM accepts a particular set of strings and nothing else, we need to prove that the TM doesn't accept any strings outside of that set.

\[ L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \]
\[ L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \]
\[ L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \]
\[ L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \} \]
RE: Languages with Verifiers
Given any string \( w \in L \), could you prove that \( w \in L \)?

That in turn would require us – in the general case – to run the TM on infinitely many strings to see what happens, since there's no general way to see what a TM does other than to run it.

**REG**

R

RE

ALL

\( L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \)

\( L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \)

\( L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \)

\( L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \} \)

R: Languages with Deciders
In addition to the RE requirements, given any string \( w \notin L \), could you prove that \( w \notin L \)?
So at least, intuitively, this doesn't seem like it's going to be possible to do. Even if we know that TM accepts exactly two strings, it's unclear how we'd prove that to someone.

\[ L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \]
\[ L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \]
\[ L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \]
\[ L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \} \]
This gives us some justification to guess that this language is probably not going to be in RE.

\[ L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \]
\[ L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \]
\[ L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \]
\[ L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \} \]
So there you have it – this language is not even in RE.

---

**L**₁ = \{ \langle M \rangle | \text{ } M \text{ is a TM and } |𝓔(M)| \geq 2 \}  

**L**₂ = \{ \langle M \rangle | \text{ } M \text{ is a TM and } |𝓔(M)| = 2 \}  

**L**₃ = \{ a^n b^n | n \in \mathbb{N} \text{ and } n > 1000 \}  

**L**₄ = \{ a^n b^n | n \in \mathbb{N} \text{ and } n \leq 1000 \}  

---

**R**: Languages with Deciders  
In addition to the **RE** requirements, given any string \( w \notin L \), could you prove that \( w \notin L \)?

---

**RE**: Languages with Verifiers  
Given any string \( w \in L \), could you prove that \( w \in L \)?
That might seem pretty surprising, given how similar this language looks to \( L_1 \).

That might seem pretty surprising, given how similar this language looks to \( L_1 \).
I chose this particular example because it highlights a key point when thinking about languages: don't try to place a language in the diagram just based on its description.

**REG**

**R**

**RE**

**ALL**

### Languages with Verifiers (RE)

Given any string $w \in L$, could you prove that $w \in L$?

### Languages with Deciders (R)

In addition to the RE requirements, given any string $w \notin L$, could you prove that $w \notin L$?

#### Languages

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$

$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$

$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
To figure out where something goes, you need to think about in terms of provability. Ultimately, it's this - rather than the way it's written - that makes things hard.

**R**: Languages with Deciders

In addition to the **RE** requirements, given any string $w \notin L$, could you prove that $w \notin L$?

**RE**: Languages with Verifiers

Given any string $w \in L$, could you prove that $w \in L$?

$L_1 = \{ \langle M \rangle | M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle | M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$

$L_3 = \{ a^n b^n | n \in \mathbb{N} \text{ and } n > 1000 \}$

$L_4 = \{ a^n b^n | n \in \mathbb{N} \text{ and } n \leq 1000 \}$
With that said, let's go take a look at the next language in our list.

\begin{align*}
L_1 &= \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \\
L_2 &= \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \\
L_3 &= \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \\
L_4 &= \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}
\end{align*}
As before, we'll start by placing it outside of \textsf{RE} and try to think about pushing it as far down as possible.

\textbf{RE: Languages with Verifiers}

Given any string $w \in L$, could you prove that $w \in L$?

\textbf{R: Languages with Deciders}

In addition to the \textsf{RE} requirements, given any string $w \notin L$, could you prove that $w \notin L$?

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$

$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$

$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
As before, we first ask whether this language happens to be in RE.

**REG**

**R**

**RE**

**ALL**

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$

$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$

$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$

**RE: Languages with Verifiers**

Given any string $w \in L$, could you prove that $w \in L$?

**R: Languages with Deciders**

In addition to the RE requirements, given any string $w \notin L$, could you prove that $w \notin L$?
So let's imagine we have an arbitrary string from this language.

**RE: Languages with Verifiers**

Given any string $w \in L$, could you prove that $w \in L$?

**R: Languages with Deciders**

In addition to the RE requirements, given any string $w \notin L$, could you prove that $w \notin L$?

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$

$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$

$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
That means that we have a string of the form $a^n b^n$ with at least 2,002 characters in it (at least 1,001 $a$'s and at least 1,001 $b$'s.)
So – given that string, could we prove to someone that the string was indeed in the language?

**R**: Languages with Deciders

In addition to the **RE** requirements, given any string \( w \notin L \), could you **prove** that \( w \notin L \)?

\[
L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |L(M)| \geq 2 \} \\
L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |L(M)| = 2 \} \\
L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \\
L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}
\]
Sure! We could just count up the a's, count up the b's, show that there are the same number, and show that there's at least 1,000.

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$
$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$
$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$
$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
Next, let's ask the follow-up question to see if $L_3$ is in $R$. If we had a string not in the language, could we prove it?
There are a lot of cases to check if the string ends up not being in the language.

**R**: Languages with Deciders

In addition to the **RE** requirements, given any string $w \notin L$, could you prove that $w \notin L$?

**RE**: Languages with Verifiers

Given any string $w \in L$, could you prove that $w \in L$?

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} $

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} $

$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} $

$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \} $
It could not have the form \(a^n b^n\), or it could have too few a's and b's in it, for example.

**L_1** = \{ \langle M \rangle | M is a TM and \(|\mathcal{L}(M)| \geq 2\) \}

**L_2** = \{ \langle M \rangle | M is a TM and \(|\mathcal{L}(M)| = 2\) \}

**L_3** = \{ \(a^n b^n\) | \(n \in \mathbb{N}\) and \(n > 1000\) \}

**L_4** = \{ \(a^n b^n\) | \(n \in \mathbb{N}\) and \(n \leq 1000\) \}
However, all of those cases are really easy to check. We either show that it has the wrong form or show that it doesn't have enough characters in it.

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \quad \text{RE: Languages with Verifiers}

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \quad \text{RE: Languages with Verifiers}

$L_3 = \{ a^nb^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \quad \text{R: Languages with Deciders}

$L_4 = \{ a^nb^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \} \quad \text{R: Languages with Deciders}
Okay, things are looking good here!
We know that this language is decidable.
As our final step, we need to ask whether or not it's regular.

\[
\begin{align*}
L_1 &= \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \\
L_2 &= \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \\
L_3 &= \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \\
L_4 &= \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}
\end{align*}
\]
REG

R

RE

ALL

So what exactly makes a language regular?

RE: Languages with Verifiers
Given any string \( w \in L \), could you prove that \( w \in L \)?

R: Languages with Deciders
In addition to the RE requirements, given any string \( w \notin L \), could you prove that \( w \notin L \)?

\[
\begin{align*}
L_1 &= \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \\
L_2 &= \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \\
L_3 &= \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \\
L_4 &= \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}
\end{align*}
\]
We have a ton of different definitions for regular languages – they're the languages of DFAs, NFAs, and regexes.

\[
L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \\
L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \\
L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \\
L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}
\]
But, as with R and RE, I think there’s a much better intuition to have about the regular languages that makes it easier to see whether something is regular.

\[
L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \\
L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \\
L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \\
L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}
\]
Specifically, the regular languages really correspond to problems that you can solve in finite memory. (This is the same intuition we used to find nonregular languages for the first time.)

REG: Problems Solvable with Finite Memory

RE: Languages with Verifiers
Given any string \( w \in L \), could you prove that \( w \in L \)?

\[ L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \]
\[ L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \]
\[ L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \]
\[ L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \} \]
REG: Problems Solvable with Finite Memory

RE: Languages with Verifiers
Given any string $w \in L$, could you prove that $w \in L$?

R: Languages with Deciders
In addition to the RE requirements, given any string $w \notin L$, could you prove that $w \notin L$?

If you're trying to determine whether a decidable language happens to be regular, think about how much information you need to remember about the input string.

$L_1 = \{ \langle M \rangle | M \text{ is a TM and } |L(M)| \geq 2 \}$
$L_2 = \{ \langle M \rangle | M \text{ is a TM and } |L(M)| = 2 \}$
$L_3 = \{ a^n b^n | n \in \mathbb{N} \text{ and } n > 1000 \}$
$L_4 = \{ a^n b^n | n \in \mathbb{N} \text{ and } n \leq 1000 \}$
REG: Problems Solvable with Finite Memory
Are there are finitely many cases to check?

RE: Languages with Verifiers
Given any string \( w \in L \), could you prove that \( w \in L \)?

R: Languages with Deciders
In addition to the RE requirements, given any string \( w \notin L \), could you prove that \( w \notin L \)?

If you only need to remember one of finitely many pieces of information, then the language is almost certainly regular, even if you can't envision a clean DFA or regex for it.

\[
\begin{align*}
L_1 &= \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \\
L_2 &= \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \\
L_3 &= \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \\
L_4 &= \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}
\end{align*}
\]
REG: Problems Solvable with Finite Memory
Are there finitely many cases to check?

RE: Languages with Verifiers
Given any string \( w \in L \), could you prove that \( w \in L \)?

R: Languages with Deciders
In addition to the RE requirements, given any string \( w \notin L \), could you prove that \( w \notin L \)?

\[
L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |L(M)| \geq 2 \}
\]
\[
L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |L(M)| = 2 \}
\]
\[
L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}
\]
\[
L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}
\]

So let's think about this here. What information do we need to keep track of?
REG: Problems Solvable with Finite Memory
Are there are finitely many cases to check?

RE: Languages with Verifiers
Given any string $w \in L$, could you prove that $w \in L$?

R: Languages with Deciders
In addition to the RE requirements, given any string $w \notin L$, could you prove that $w \notin L$?

Fundamentally, we'd have to keep track of how many a's we've seen, since if we can't do that, we can't match it against the number of b's.

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$

$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$

$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
REG: Problems Solvable with Finite Memory
Are there are finitely many cases to check?

RE: Languages with Verifiers
Given any string \( w \in L \), could you prove that \( w \in L \)?

R: Languages with Deciders
In addition to the RE requirements, given any string \( w \notin L \), could you prove that \( w \notin L \)?

\[ L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \]
\[ L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \]
\[ L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \]
\[ L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \} \]

That's a problem: there are infinitely many possible choices for the number of a's that we'd have to remember, and we can't remember which number we've seen with finitely many states!
So this gives us the intuition that $L_3$ is almost certainly going to be nonregular.

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$

$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$

$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
You can formally prove this by using the Myhill-Nerode theorem. I highly recommend it – it’s good practice!

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$
$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$
$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$
$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
REG: Problems Solvable with Finite Memory

Are there finitely many cases to check?

RE: Languages with Verifiers

Given any string $w \in L$, could you prove that $w \in L$?

R: Languages with Deciders

In addition to the RE requirements, given any string $w \notin L$, could you prove that $w \notin L$?

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$

$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$

$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
We've seen how to use our key intuition for regular languages – they're languages you can solve in finite space – to check whether something is regular.

REG: Problems Solvable with Finite Memory
Are there are finitely many cases to check?

RE: Languages with Verifiers
Given any string $w \in L$, could you prove that $w \in L$?

R: Languages with Deciders
In addition to the RE requirements, given any string $w \notin L$, could you prove that $w \notin L$?

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$

$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$

$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
With all that said and done, let's move on to our last language here.

### REG: Problems Solvable with Finite Memory
Are there are finitely many cases to check?

### RE: Languages with Verifiers
Given any string $w \in L$, could you prove that $w \in L$?

### R: Languages with Deciders
In addition to the RE requirements, given any string $w \notin L$, could you prove that $w \notin L$?

$L_1 = \{ \langle M \rangle | M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle | M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$

$L_3 = \{ a^n b^n | n \in \mathbb{N} \text{ and } n > 1000 \}$

$L_4 = \{ a^n b^n | n \in \mathbb{N} \text{ and } n \leq 1000 \}$
While normally we've talked about starting from the outside and moving inward, for this language I think you can probably see that this is going to be decidable, so let's start it there.

\[ L_1 = \{ \langle M \rangle | M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \]
\[ L_2 = \{ \langle M \rangle | M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \]
\[ L_3 = \{ a^n b^n | n \in \mathbb{N} \text{ and } n > 1000 \} \]
\[ L_4 = \{ a^n b^n | n \in \mathbb{N} \text{ and } n \leq 1000 \} \]
The question now is whether it's regular or not.

\[ L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |L(M)| \geq 2 \} \]
\[ L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |L(M)| = 2 \} \]
\[ L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \]
\[ L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \} \]
The answer is yes. Here's a number of different ways to think about why.

### Languages with Verifiers (RE)

Given any string \( w \in L \), could you **prove** that \( w \in L \)?

### Languages with Deciders (R)

In addition to the RE requirements, given any string \( w \notin L \), could you **prove** that \( w \notin L \)?

### Problems Solvable with Finite Memory (REG)

Are there finitely many cases to check?

---

\[
L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \\
L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \\
L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \\
L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}
\]
First, we can think about this from an information perspective. To check whether a string is in this language, we need to keep track of how many $a$'s there are and how many $b$'s there are...

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$
$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$
$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$
$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
REG: Problems Solvable with Finite Memory

Are there are finitely many cases to check?

RE: Languages with Verifiers

Given any string \( w \in L \), could you prove that \( w \in L \)?

R: Languages with Deciders

In addition to the RE requirements, given any string \( w \notin L \), could you prove that \( w \notin L \)?

\[
\begin{align*}
L_1 &= \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \\
L_2 &= \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \\
L_3 &= \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \\
L_4 &= \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}
\end{align*}
\]

...but only up to a point. After we see 1,001 copies of either character, we know that the string isn't in the language.
This means that we just need to remember how many a's and b's we've seen (within the limits) and whether we're still reading a's or b's.

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$

$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$

$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
That means we only need a finite amount of information to decide whether a string is in the language, so using our intuition for the regular languages, this one will be regular.

\[ L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \]
\[ L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \]
\[ L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \]
\[ L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \} \]
Here's another approach we can take.

Given any string $w \in L$, could you *prove* that $w \in L$?

In addition to the RE requirements, given any string $w \notin L$, could you *prove* that $w \notin L$?

REG: Problems Solvable with Finite Memory

Are there are finitely many cases to check?

RE: Languages with Verifiers

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$

$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$

$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$

R: Languages with Deciders

In addition to the RE requirements, given any string $w \notin L$, could you *prove* that $w \notin L$?
REG: Problems Solvable with Finite Memory
Are there are finitely many cases to check?

RE: Languages with Verifiers
Given any string $w \in L$, could you prove that $w \in L$?

R: Languages with Deciders
In addition to the RE requirements, given any string $w \notin L$, could you prove that $w \notin L$?

$L_1 = \{ \langle M \rangle | M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$
$L_2 = \{ \langle M \rangle | M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$
$L_3 = \{ a^n b^n | n \in \mathbb{N} \text{ and } n > 1000 \}$
$L_4 = \{ a^n b^n | n \in \mathbb{N} \text{ and } n \leq 1000 \}$
There's only 1,001 of them, corresponding to all the different choices of n we can make.

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$

$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$

$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
REG: Problems Solvable with Finite Memory

Are there are finitely many cases to check?

RE: Languages with Verifiers

Given any string \( w \in L \), could you prove that \( w \in L \)?

R: Languages with Deciders

In addition to the RE requirements, given any string \( w \notin L \), could you prove that \( w \notin L \)?

As you proved on Problem Set 7, all finite languages are regular. That means that this language has to be regular.

\[ L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \]
\[ L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \]
\[ L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \]
\[ L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \} \]
REG: Problems Solvable with Finite Memory

Are there finitely many cases to check?

L₁ = \{ \langle M \rangle | M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} 
L₂ = \{ \langle M \rangle | M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} 
L₃ = \{ a^n b^n | n \in \mathbb{N} \text{ and } n > 1000 \} 
L₄ = \{ a^n b^n | n \in \mathbb{N} \text{ and } n \leq 1000 \}

RE: Languages with Verifiers

Given any string \( w \in L \), could you \textit{prove} that \( w \in L \)?

As a final option, we can think about this in terms of DFA or regex design.

R: Languages with Deciders

In addition to the RE requirements, given any string \( w \notin L \), could you \textit{prove} that \( w \notin L \)?
REG: Problems Solvable with Finite Memory
Are there are finitely many cases to check?

RE: Languages with Verifiers
Given any string \( w \in L \), could you *prove* that \( w \in L \)?

R: Languages with Deciders
In addition to the RE requirements, given any string \( w \notin L \), could you *prove* that \( w \notin L \)?

\[
\begin{align*}
L_1 &= \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{T}(M)| \geq 2 \} \\
L_2 &= \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{T}(M)| = 2 \} \\
L_3 &= \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \\
L_4 &= \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}
\end{align*}
\]

You could imagine building a (huge) regex for this language:
\( \varepsilon \cup ab \cup aabb \cup aaabbb \cup \ldots \cup a^{1000}b^{1000} \)
So that means that it's going to be regular.

Given any string $w \in L$, could you prove that $w \in L$?

In addition to the RE requirements, given any string $w \notin L$, could you prove that $w \notin L$?

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} 

L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}

L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}

L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}
REG: Problems Solvable with Finite Memory

Are there are finitely many cases to check?

RE: Languages with Verifiers

Given any string \( w \in L \), could you prove that \( w \in L \)?

R: Languages with Deciders

In addition to the RE requirements, given any string \( w \notin L \), could you prove that \( w \notin L \)?

By now we've successfully placed all the languages in to the Lava Diagram. Woohoo!

\[
L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}
\]
\[
L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}
\]
\[
L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}
\]
\[
L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}
\]
**REG:** Problems Solvable with Finite Memory

Are there are finitely many cases to check?

**RE:** Languages with Verifiers

Given any string \( w \in L \), could you prove that \( w \in L \)?

**R:** Languages with Deciders

In addition to the RE requirements, given any string \( w \notin L \), could you prove that \( w \notin L \)?

Let's do a quick recap of what all of the different regions mean and how best to think about them.

\[
L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}
\]

\[
L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}
\]

\[
L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}
\]

\[
L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}
\]
$L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$

$L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$

$L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
REG: Problems Solvable with Finite Memory
Are there are finitely many cases to check?

RE: Languages with Verifiers
Given any string \( w \in L \), could you prove that \( w \in L \)?

R: Languages with Deciders
In addition to the \( \text{RE} \) requirements, given any string \( w \notin L \), could you prove that \( w \notin L \)?

Next, the \( R \) languages. If that you already know your language is in \( \text{RE} \), you can figure out whether it's in \( R \) by asking whether, for any string not in the language, you can prove it's not in the language.

\[
L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}
\]
\[
L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}
\]
\[
L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}
\]
\[
L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}
\]
### REG: Problems Solvable with Finite Memory

**Are there are finitely many cases to check?**

### RE: Languages with Verifiers

Given any string $w \in L$, could you _prove_ that $w \in L$?

### R: Languages with Deciders

In addition to the RE requirements, given any string $w \notin L$, could you _prove_ that $w \notin L$?

---

Finally, the regular languages. Those are the ones that you can solve given only finite resources.

- $L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \}$
- $L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \}$
- $L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$
- $L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$
\[ L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \]

\[ L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \]

\[ L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \]

\[ L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \} \]
**REG: Problems Solvable with Finite Memory**

Are there finitely many cases to check?

**RE: Languages with Verifiers**

Given any string \( w \in L \), could you prove that \( w \in L \)?

**R: Languages with Deciders**

In addition to the RE requirements, given any string \( w \notin L \), could you prove that \( w \notin L \)?

---

\[ L_1 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| \geq 2 \} \]

\[ L_2 = \{ \langle M \rangle \mid M \text{ is a TM and } |\mathcal{L}(M)| = 2 \} \]

\[ L_3 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n > 1000 \} \]

\[ L_4 = \{ a^n b^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \} \]
Hope this helps!

Please feel free to ask questions if you have them.
Did you find this useful? If so, let us know! We can go and make more guides like these.