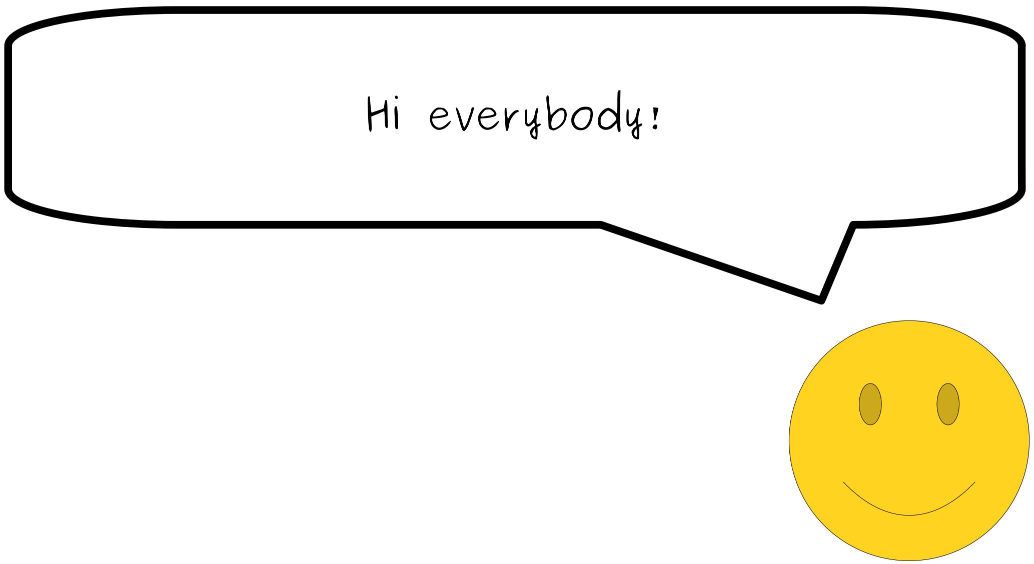
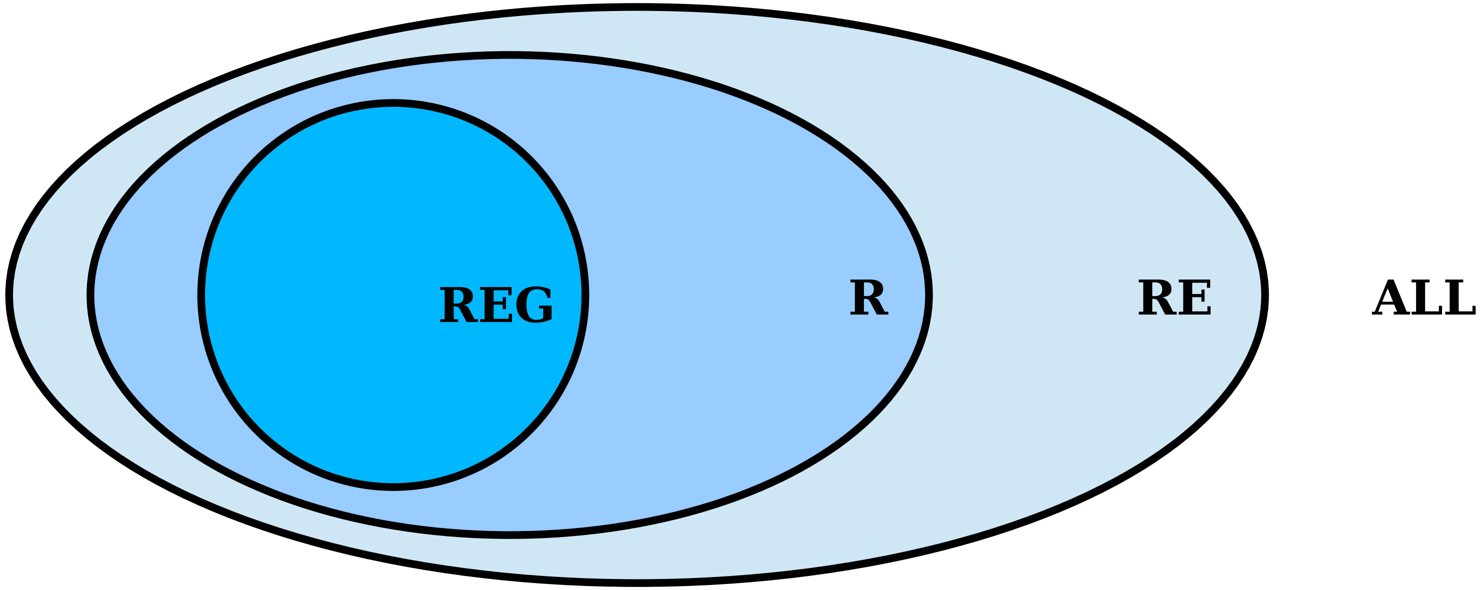
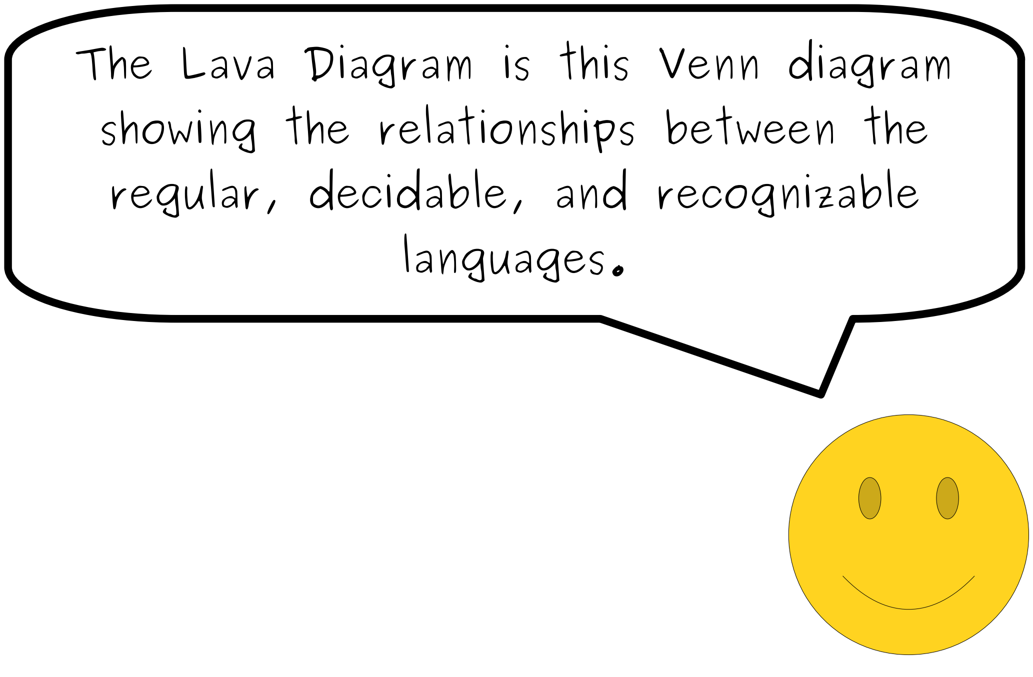
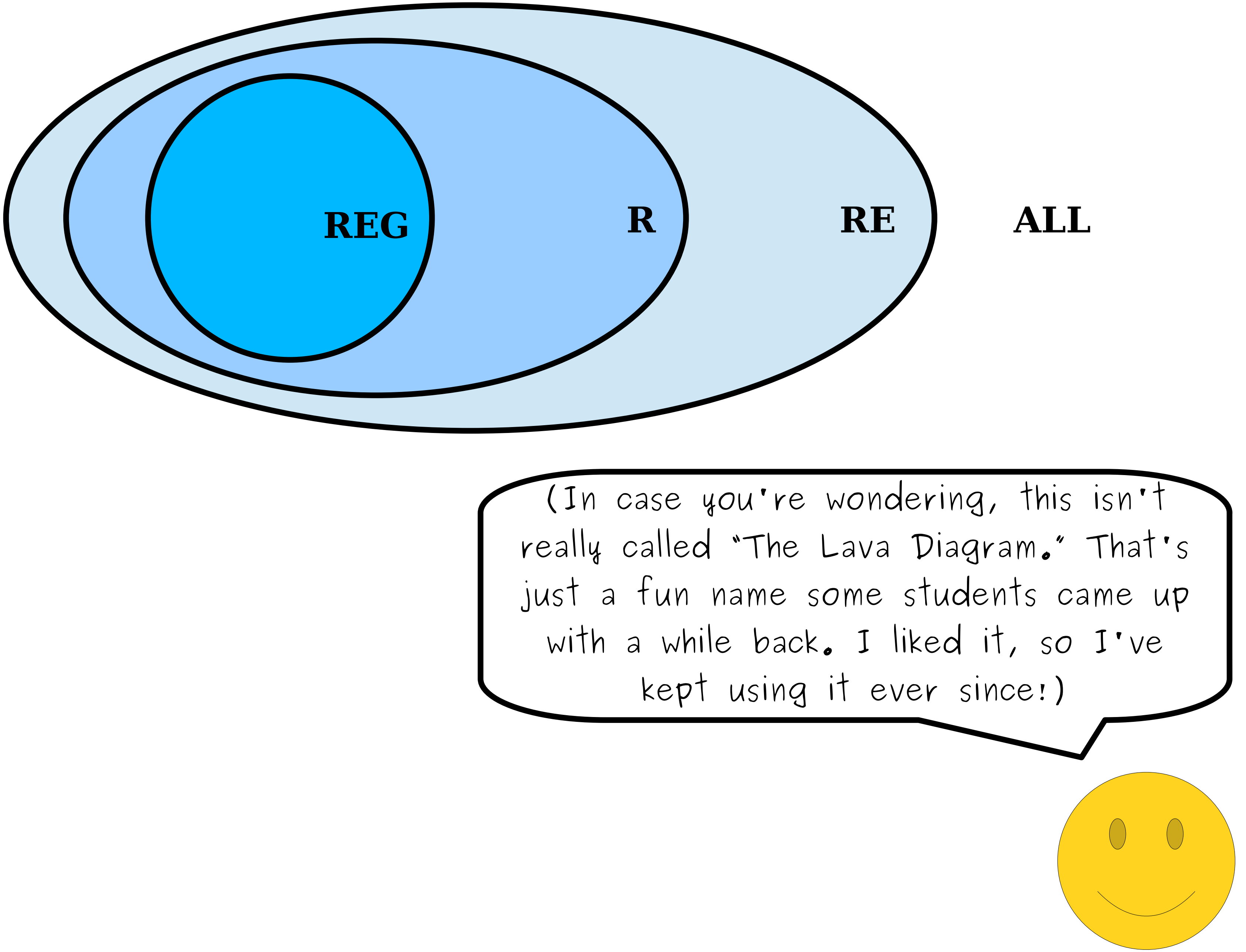
The Guide to the Lava Diagram

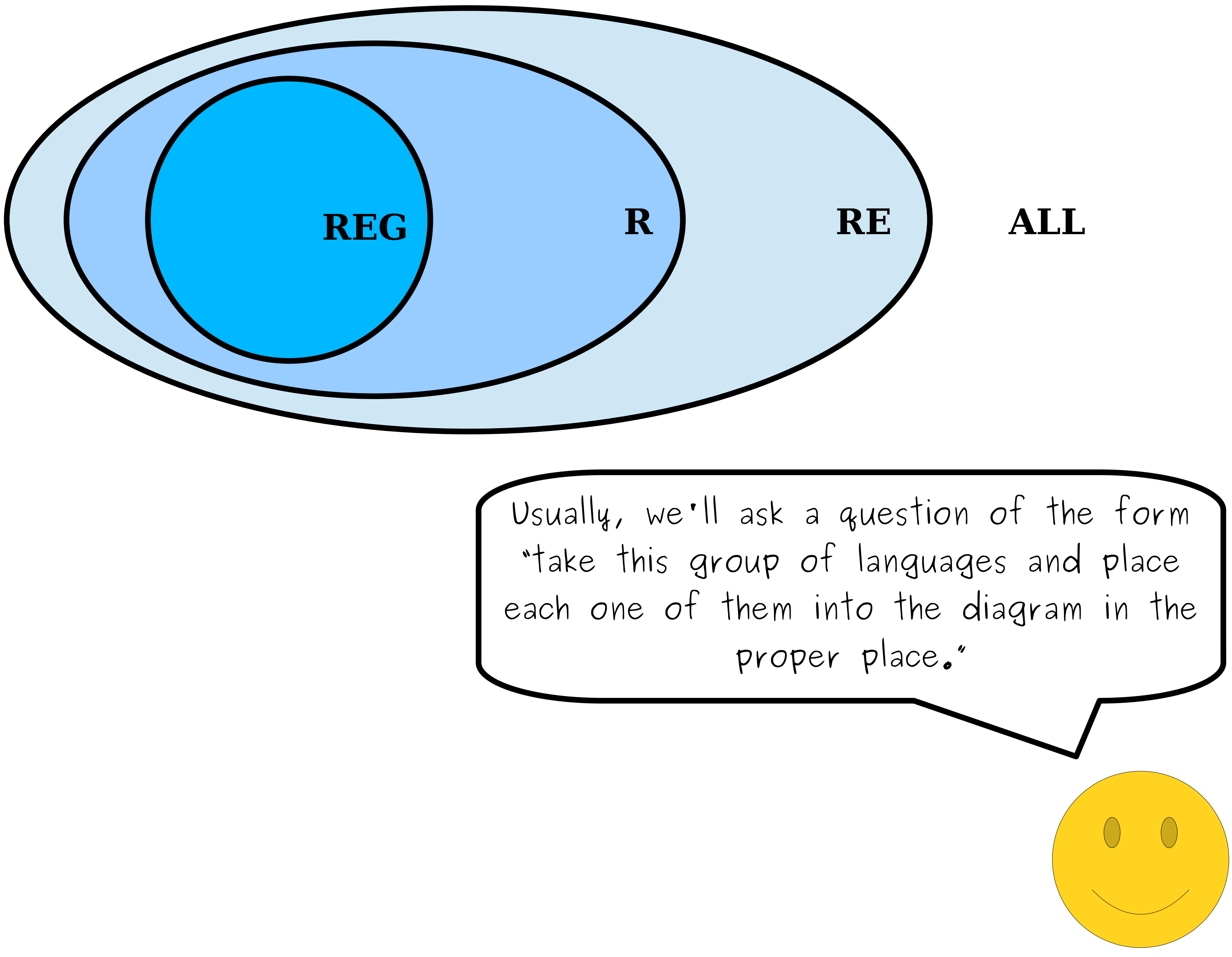

Hi everybody!

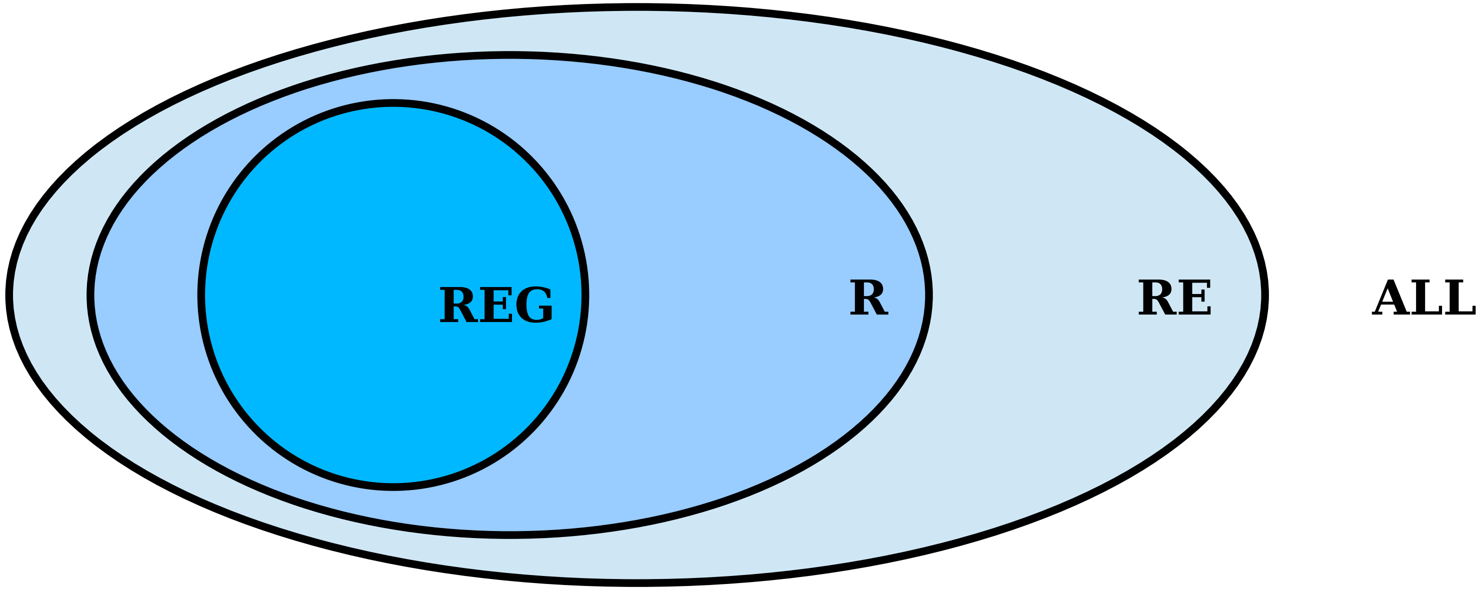
As you probably noticed on Problem Set Nine – we love asking questions about "The Lava Diagram."

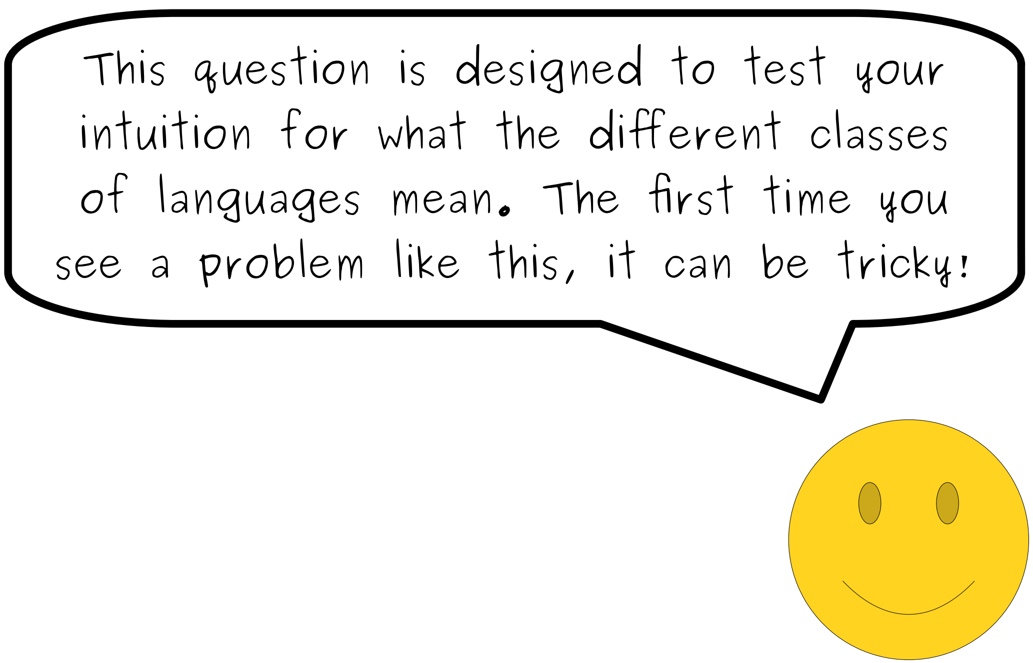

REG
R
RE
ALL

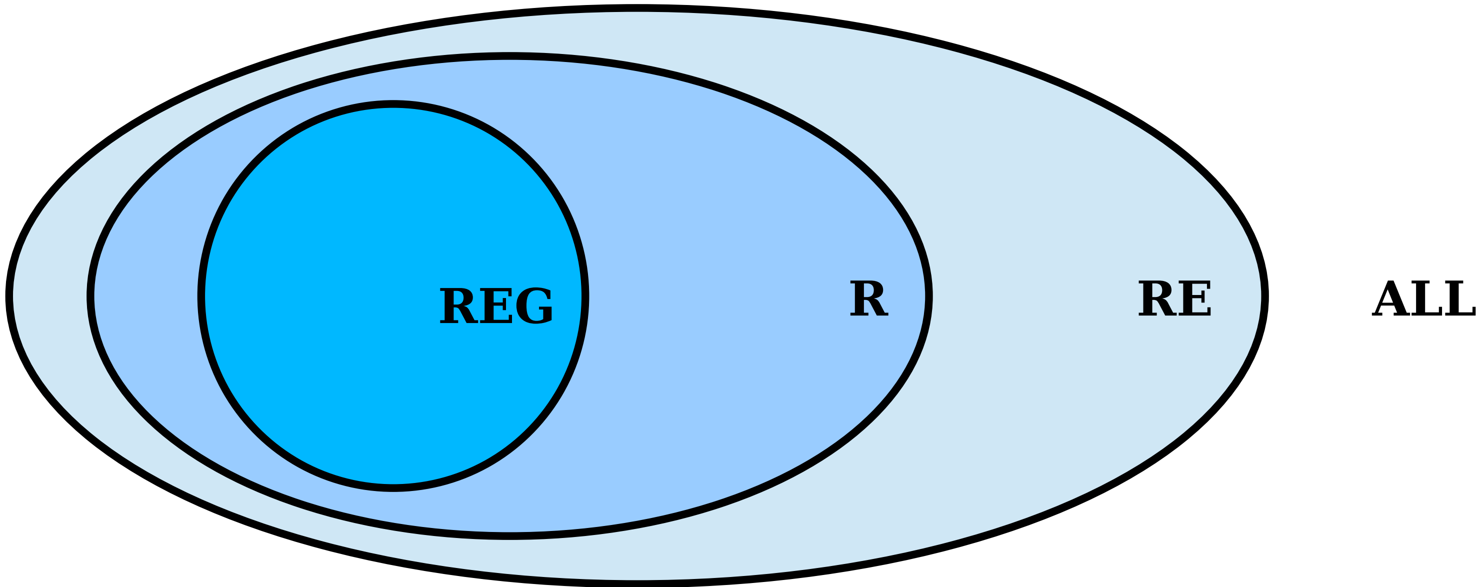

The Lava Diagram is this Venn diagram showing the relationships between the regular, decidable, and recognizable languages.


REG
R
RE
ALL
(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!)

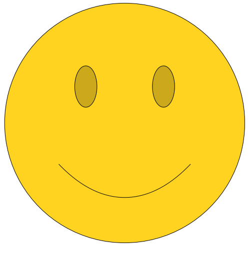

REG
R
RE
ALL
Usually, we'll ask a question of the form "take this group of languages and place each one of them into the diagram in the proper place."


REG
R
RE
ALL


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.

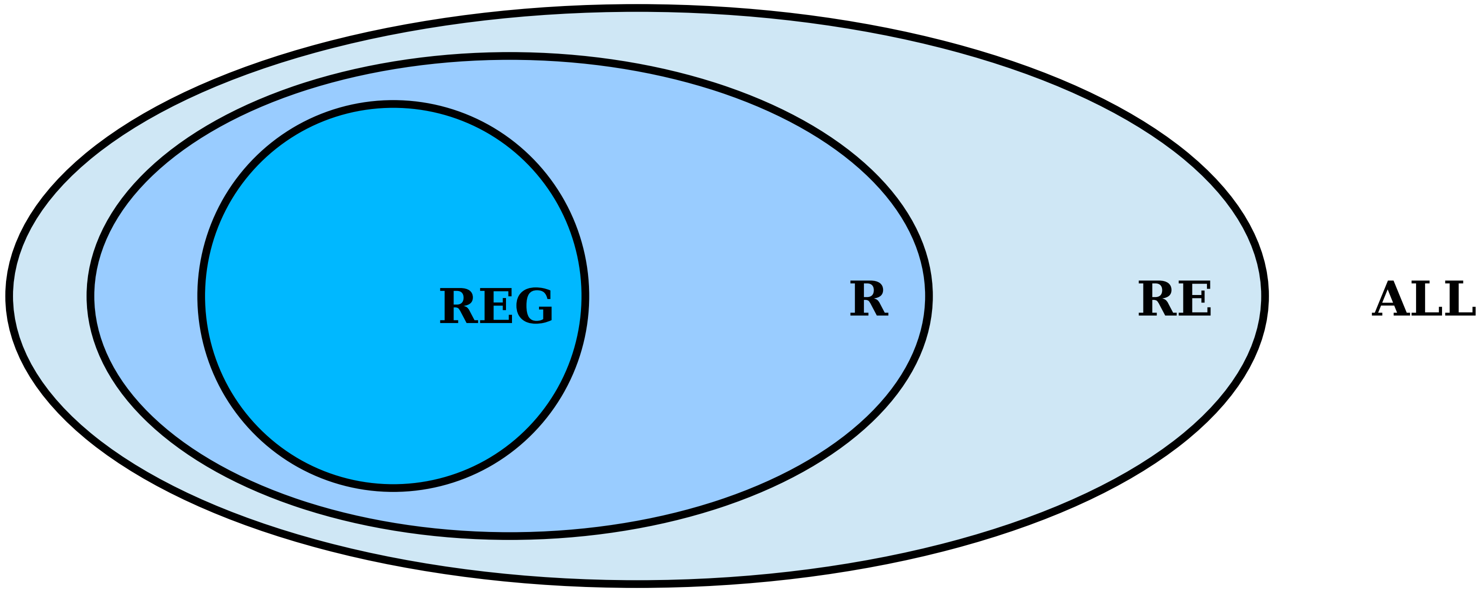


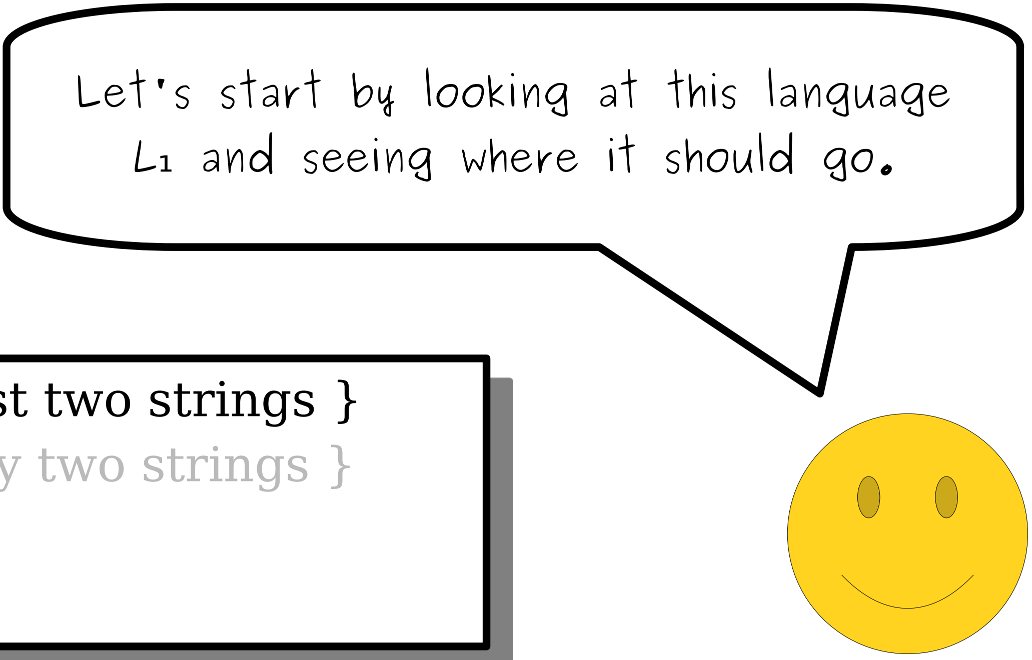
$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

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



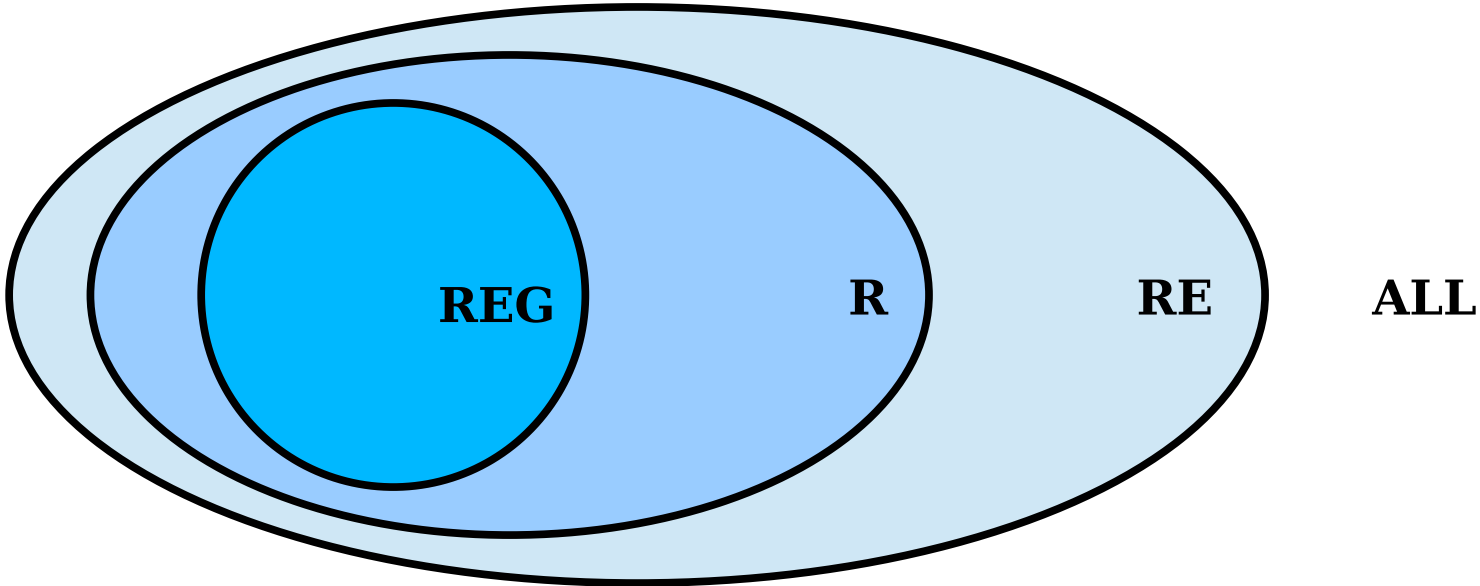


L_1 = { ⟨M⟩ | M is a TM that accepts at least two strings }

L_2 = { ⟨M⟩ | M is a TM that accepts exactly two strings }

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

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



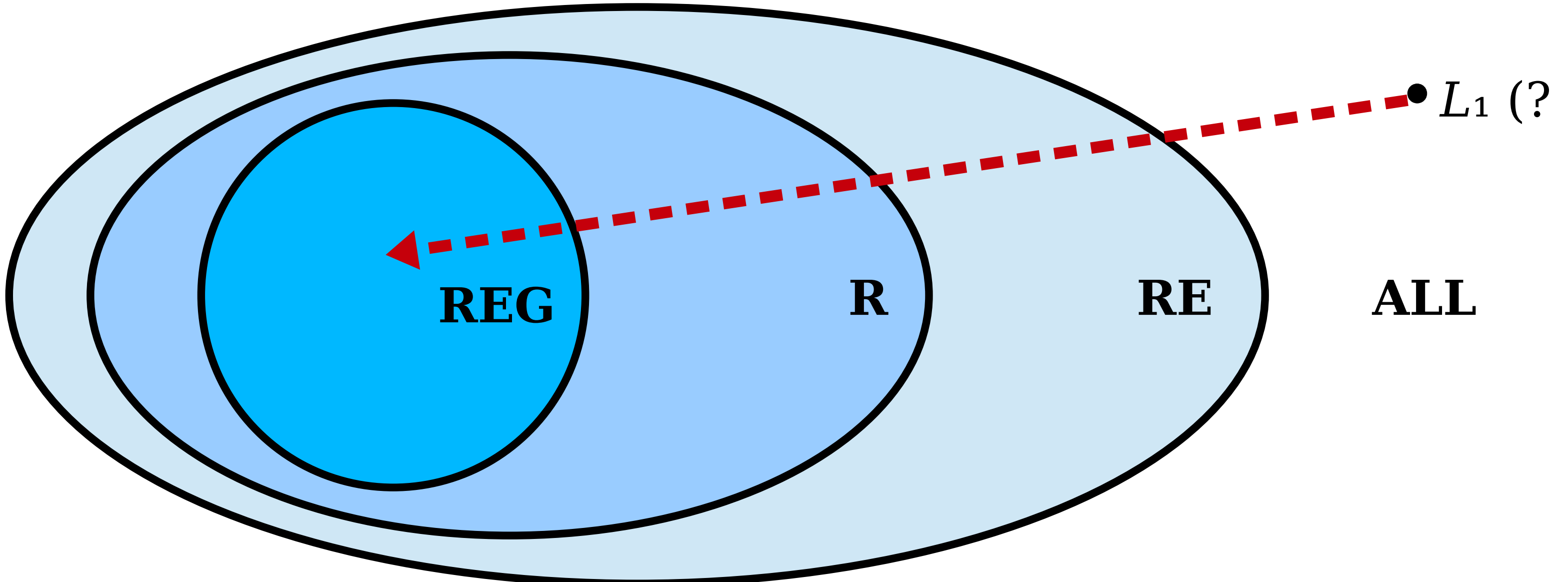
There are a couple of different strategies you can use to work through these problems, but the one we find the most useful is to start from the outside and work inward.

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

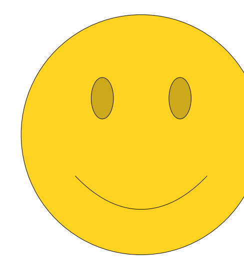
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

$L_4 = \{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N}$ 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.

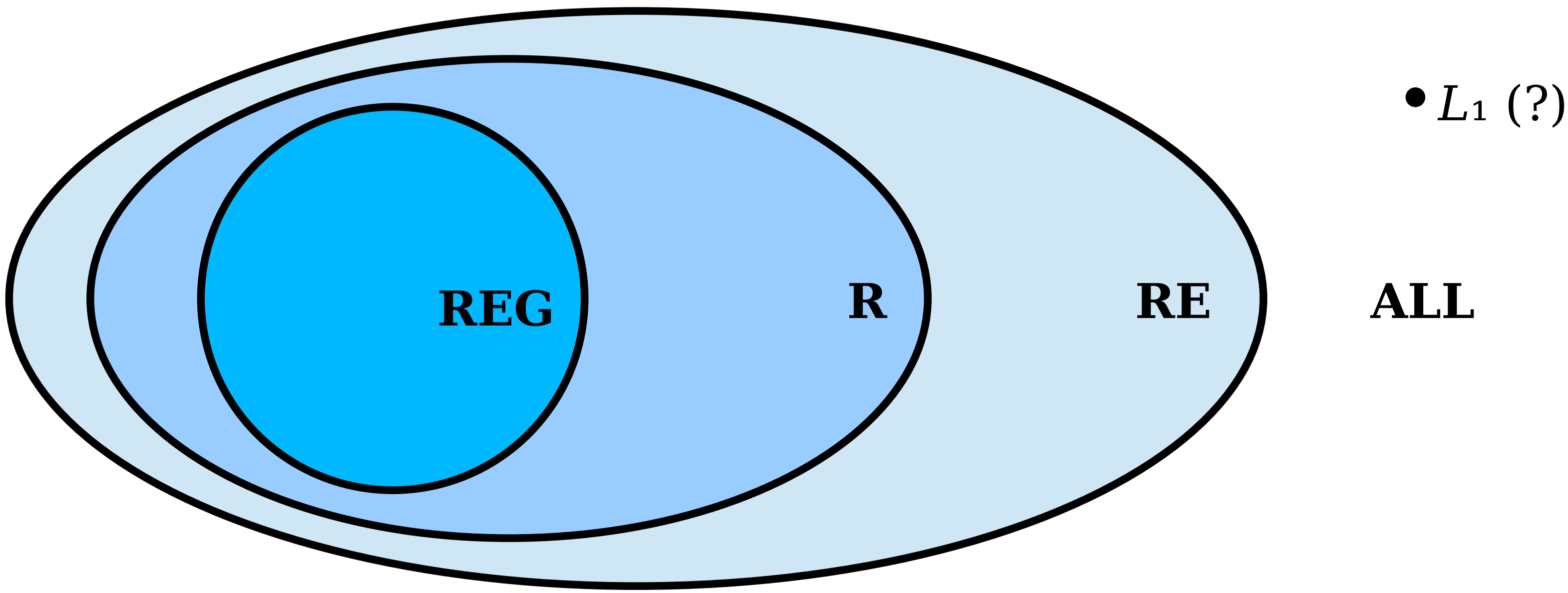


$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

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



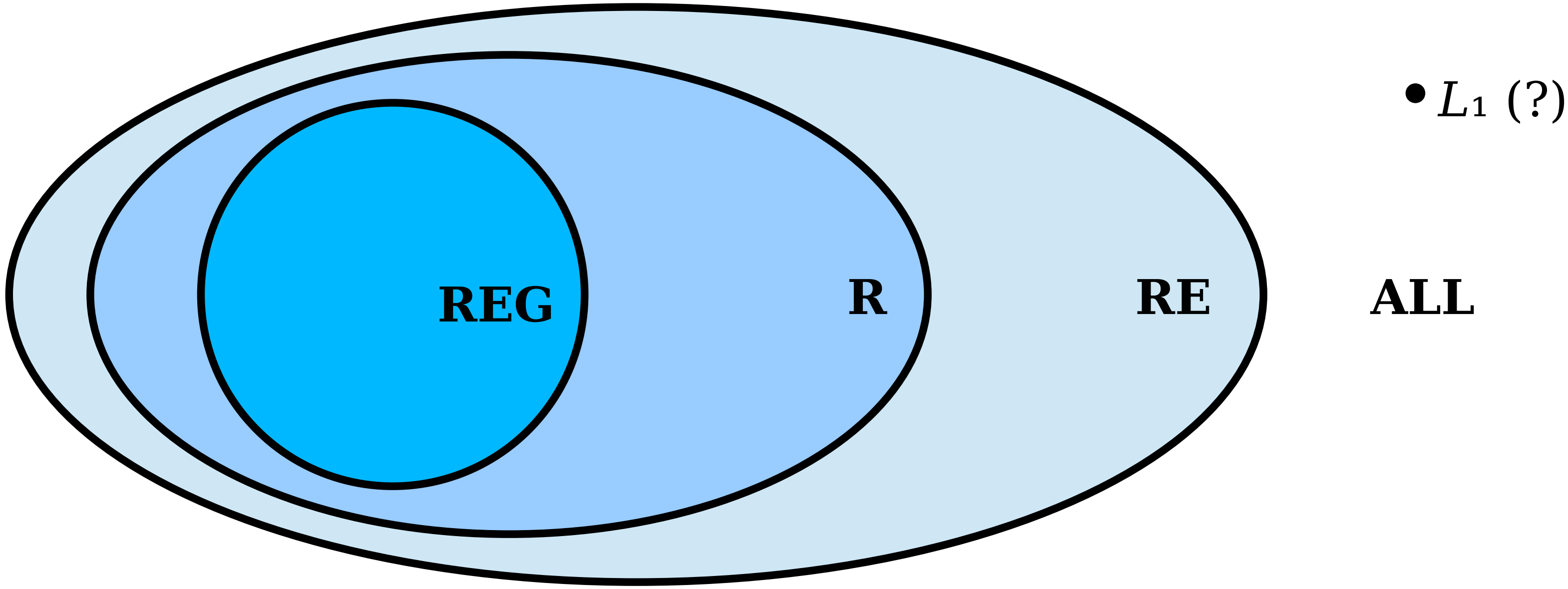
The very first question we should ask ourselves, therefore, is whether this language belongs to **RE**.

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

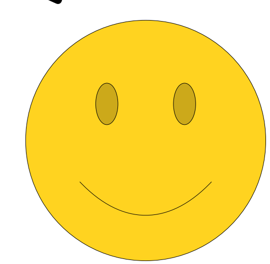
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

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



So what exactly is the class **RE**?

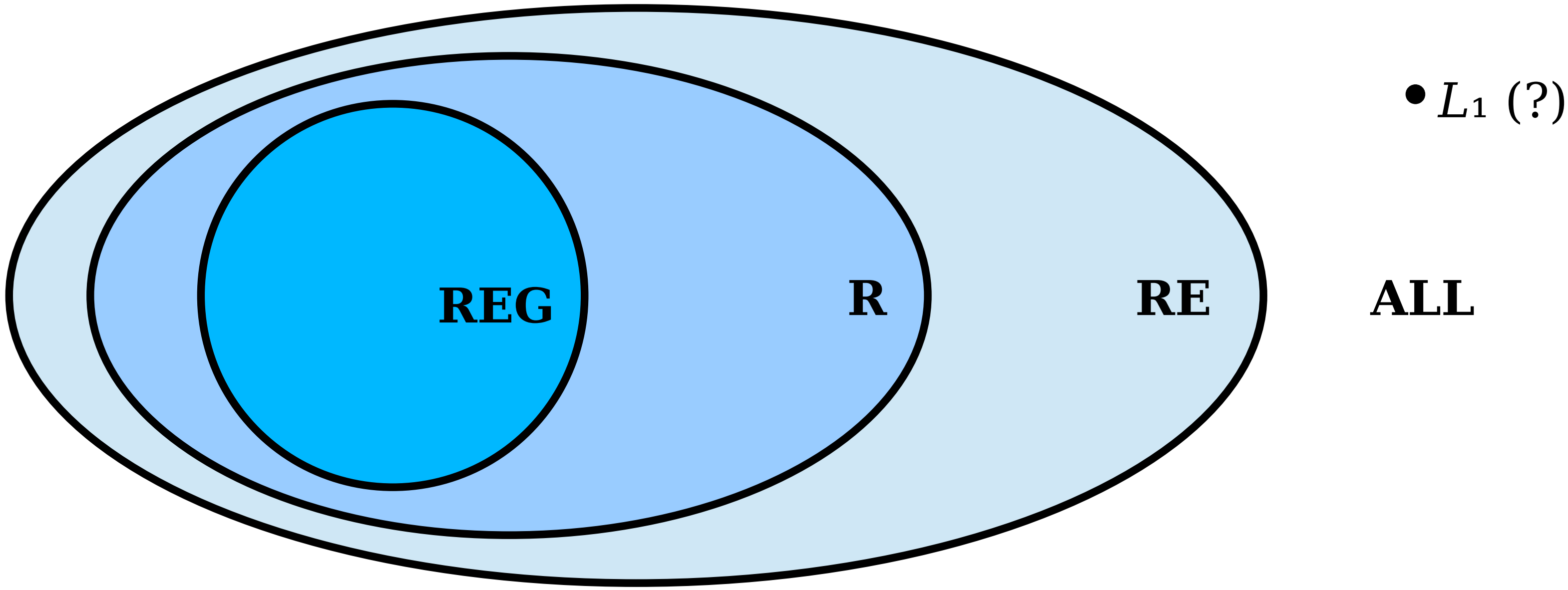


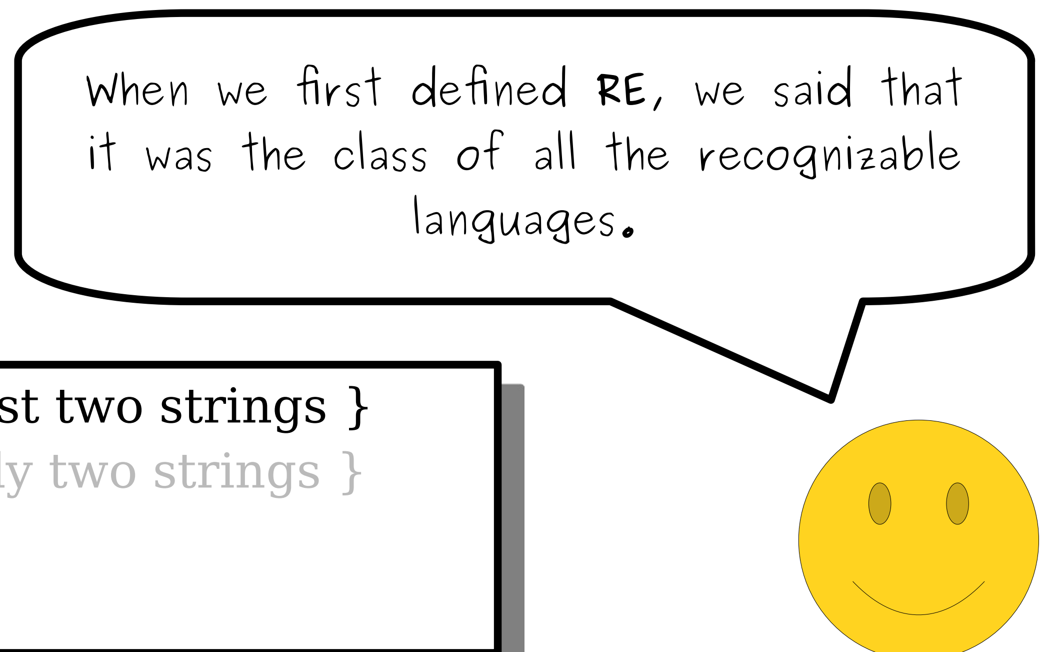
$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

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



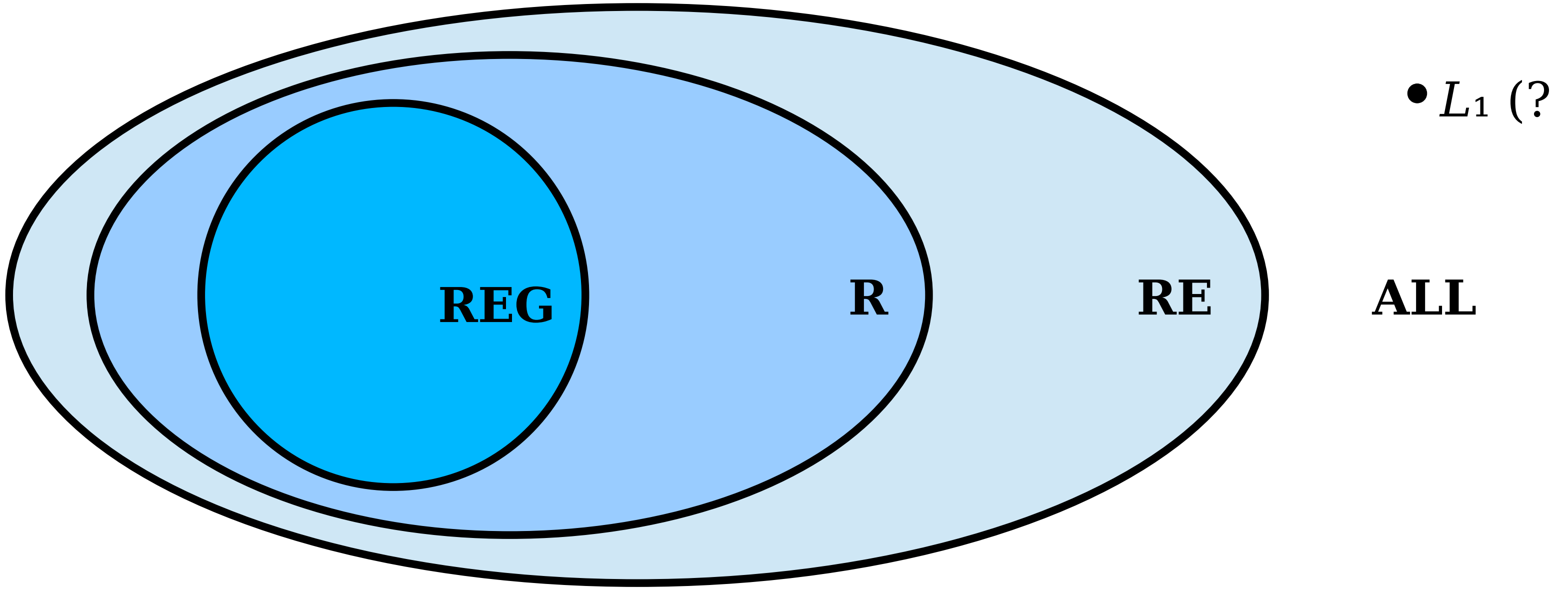


$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

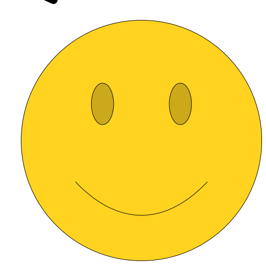
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

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



This means that we <u>could</u> try to think about **RE** as "the class of problems with recognizers."

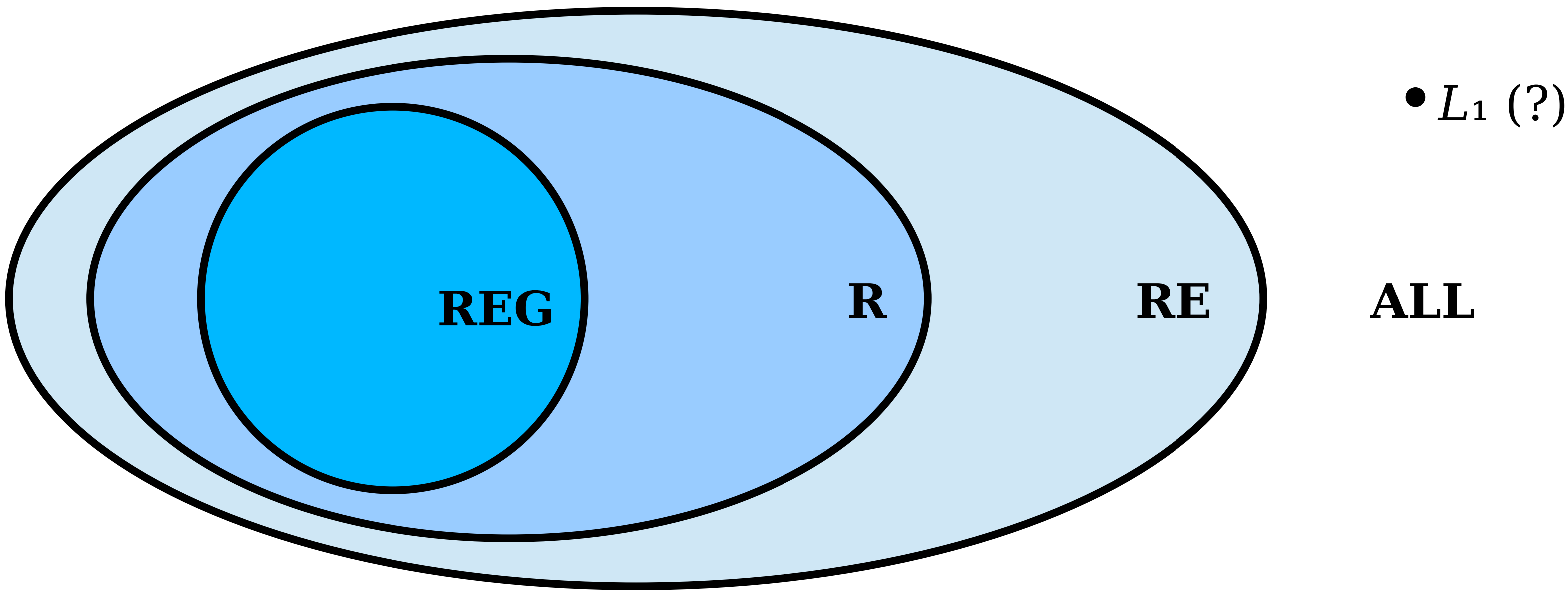


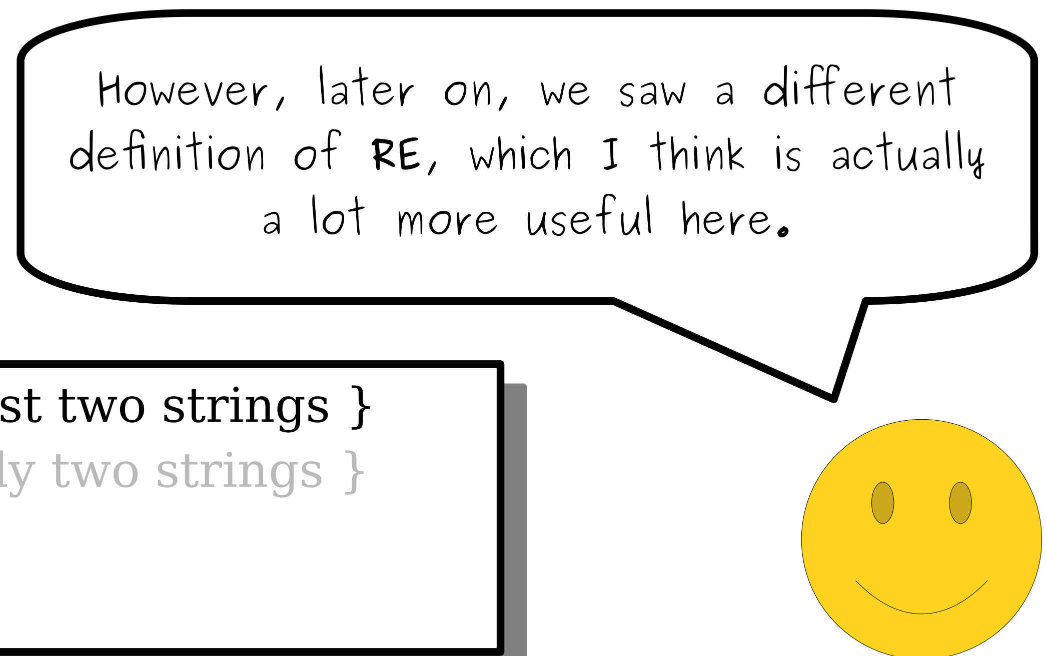
$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

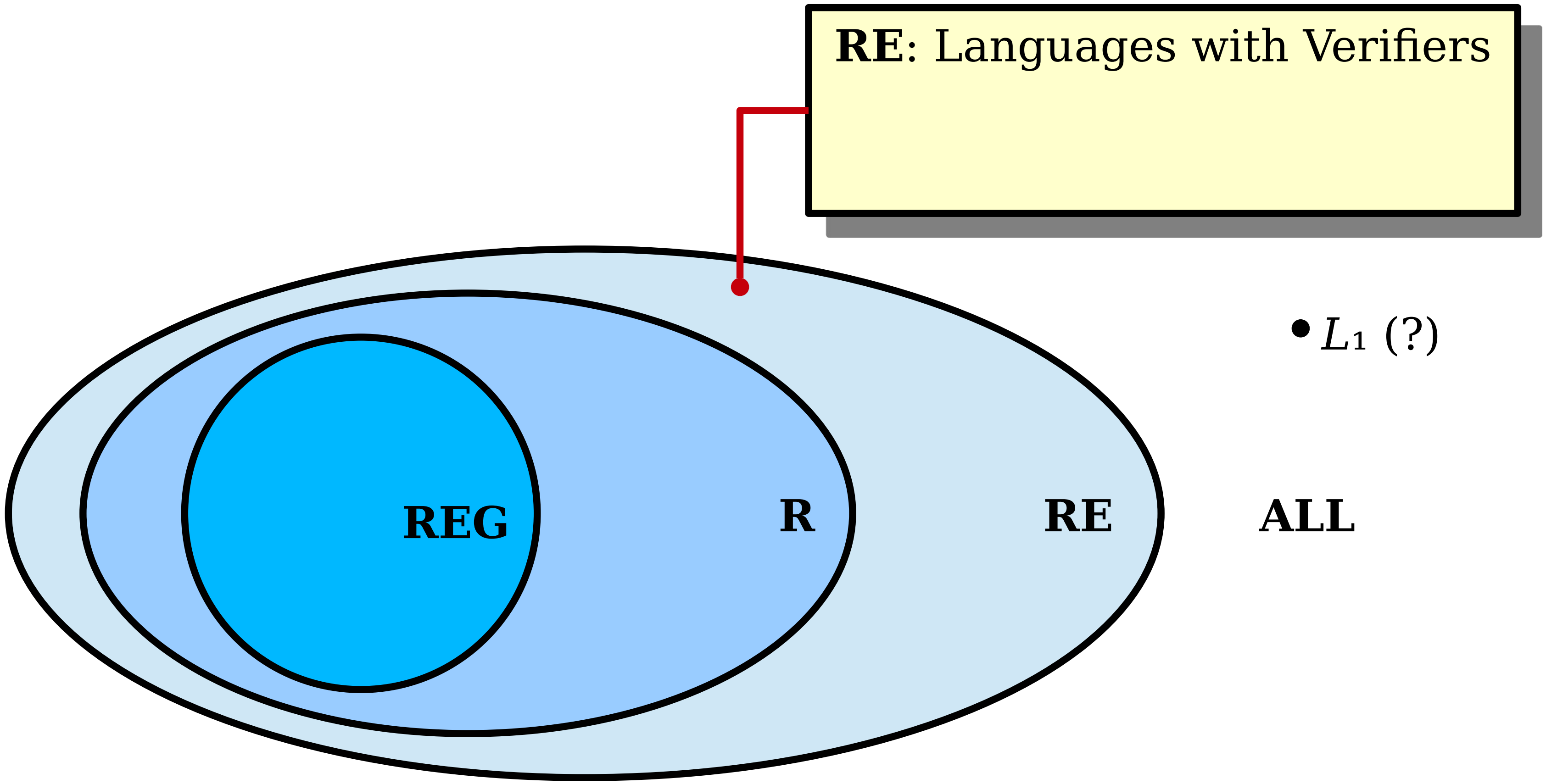
$L_3 = \{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N}$ and $n > 1000 \}$

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

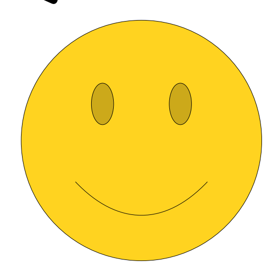




L_1 = { $\langle M \rangle$ | M is a TM that accepts at least two strings }
L_2 = { $\langle M \rangle$ | M is a TM that accepts exactly two strings }
L_3 = { $\mathbf{a}^n\mathbf{b}^n$ | $n \in \mathbb{N}$ and $n > 1000$ }
L_4 = { $\mathbf{a}^n\mathbf{b}^n$ | $n \in \mathbb{N}$ and $n \leq 1000$ }



Specifically, we saw that **RE** is the class of languages that have verifiers.



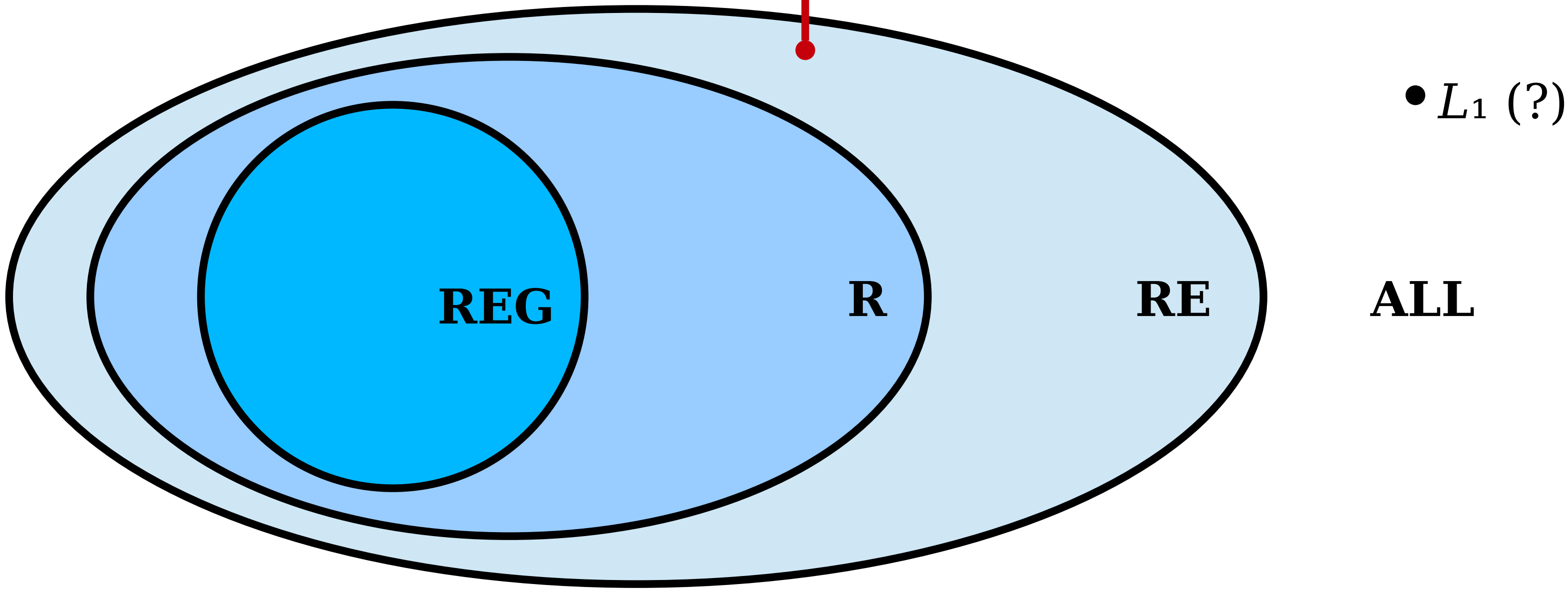
$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

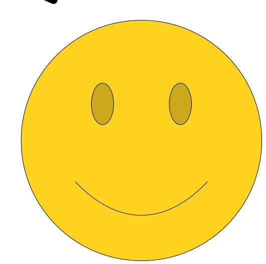
$L_3 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}$ and $n > 1000 \}$

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

RE: Languages with Verifiers



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."



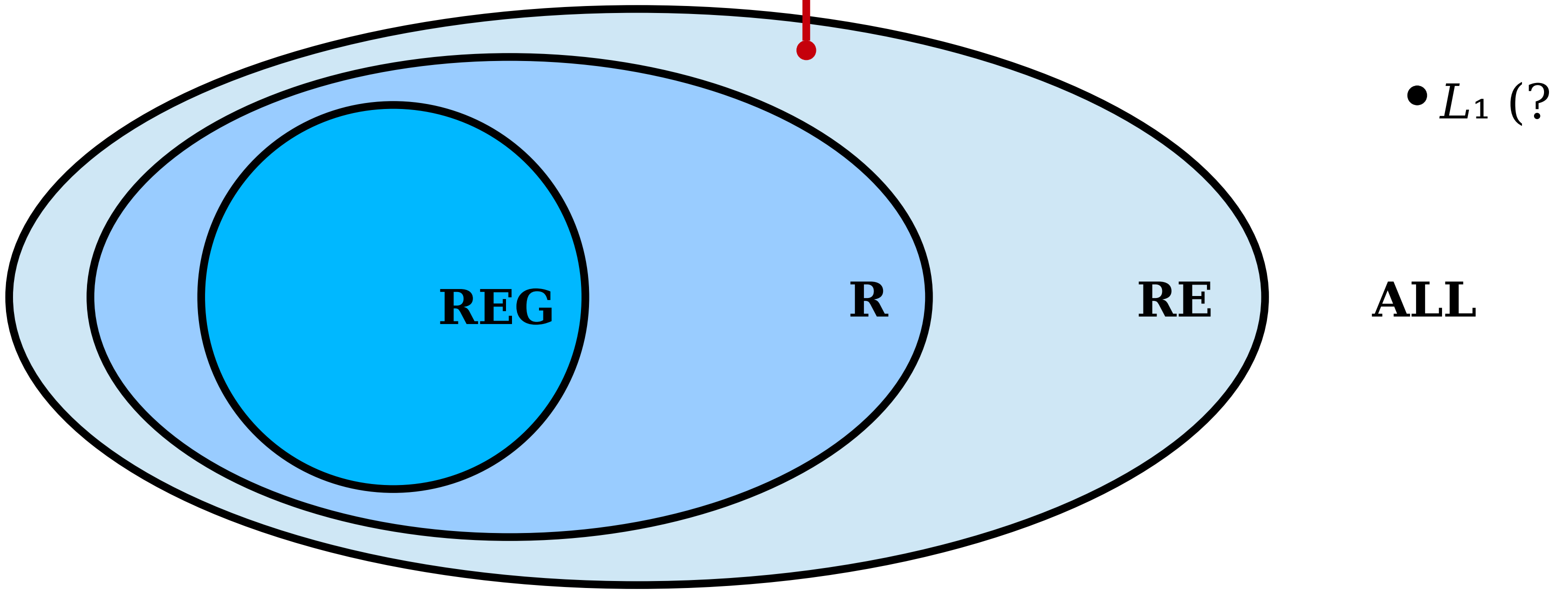
$L_1 = \{ \langle M \rangle \mid M \text{ is a TM that accepts at least two strings} \}$

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM that accepts exactly two strings} \}$

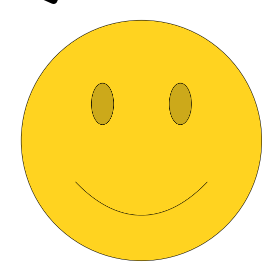
$L_3 = \{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$

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

RE: Languages with Verifiers



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.



$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

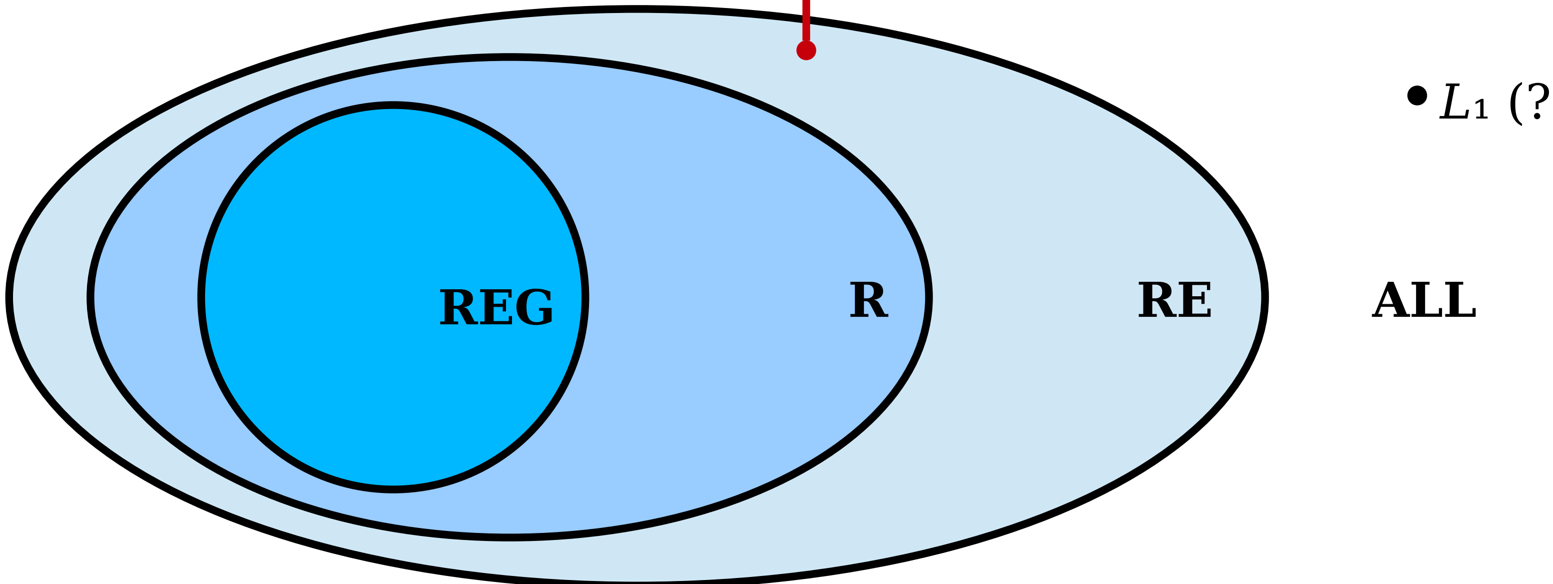
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

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

RE: Languages with Verifiers

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

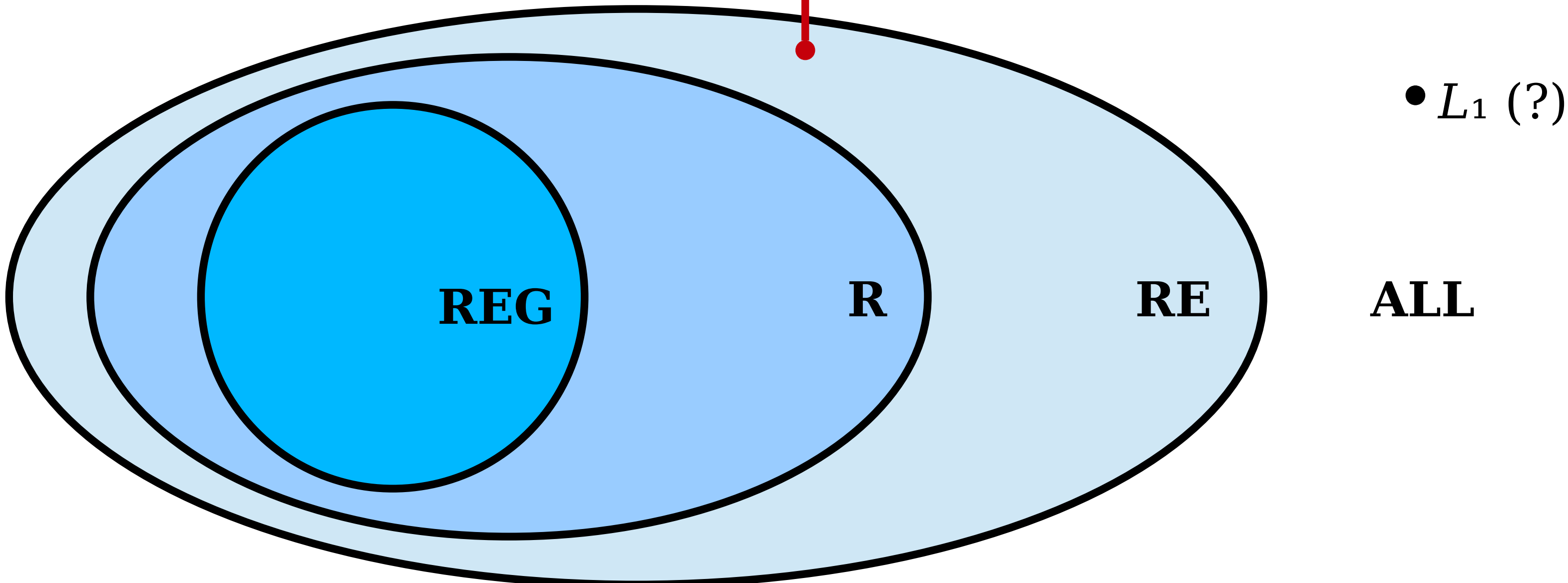


In that sense, you can think of the **RE** languages this way: they're the languages where, for any string in the language, there's some way to <u>prove</u> that the string is indeed in the language.

L_1 = { ⟨M⟩ | M is a TM that accepts at least two strings }
L_2 = { ⟨M⟩ | M is a TM that accepts exactly two strings }
L_3 = { $\mathtt{a}^n\mathtt{b}^n$ | $n \in \mathbb{N}$ and $n > 1000$ }
L_4 = { $\mathtt{a}^n\mathtt{b}^n$ | $n \in \mathbb{N}$ and $n \leq 1000$ }

RE: Languages with Verifiers

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



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.

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

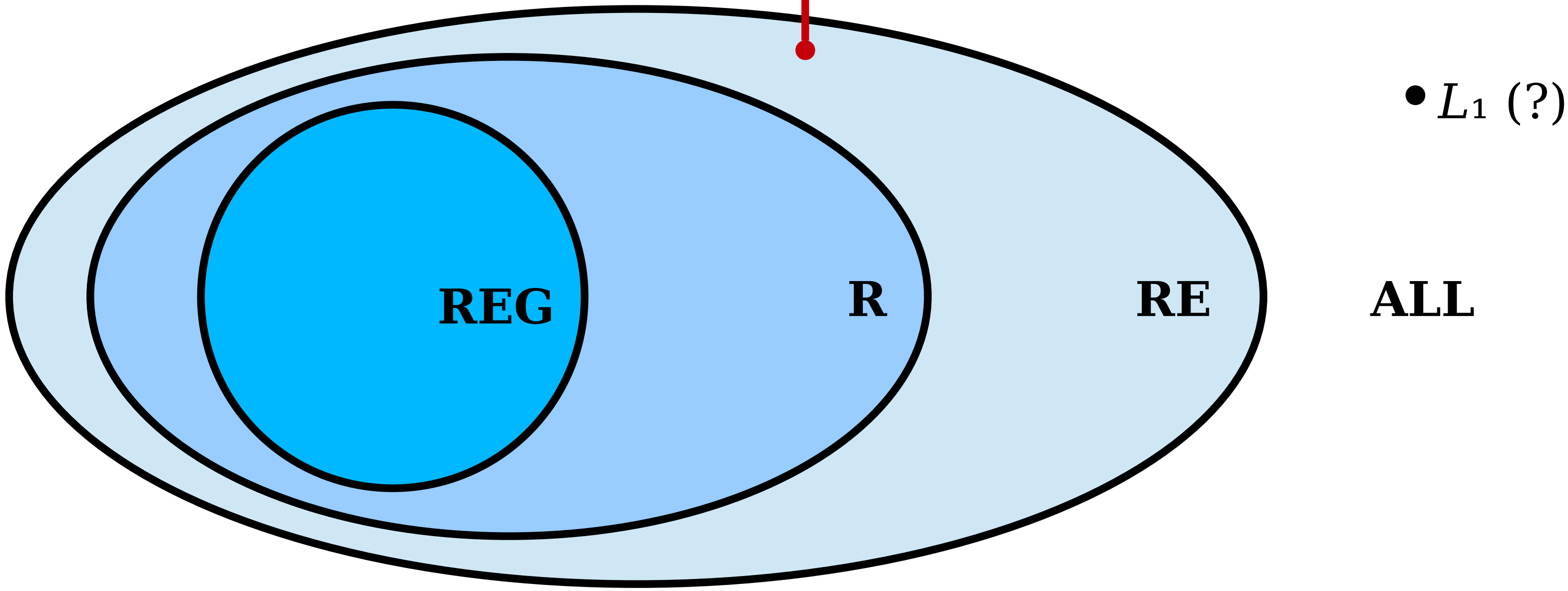
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

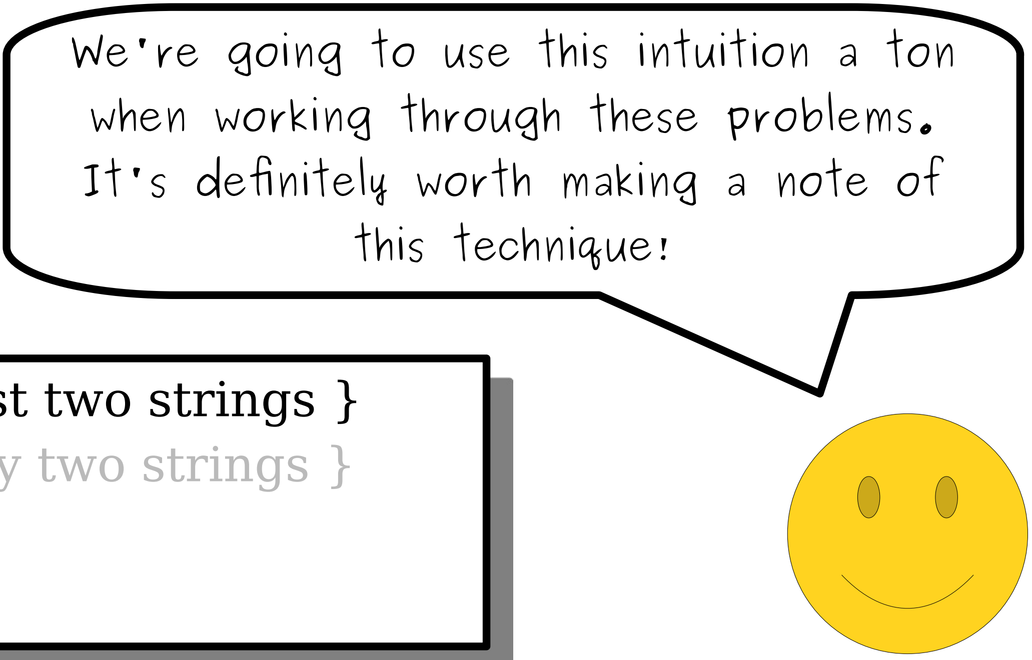
$L_3 = \{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N}$ and $n > 1000 \}$

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

RE: Languages with Verifiers

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





$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

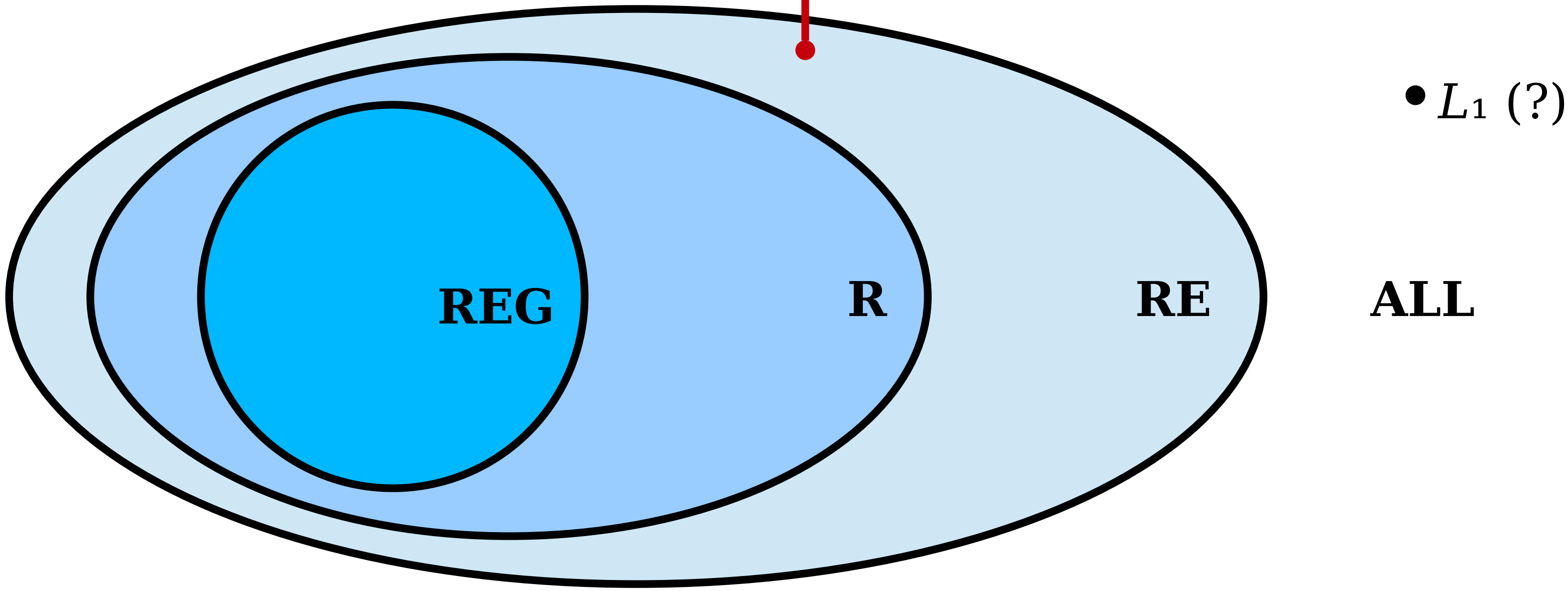
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

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

RE: Languages with Verifiers

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



So let's go focus our attention to the particular language L_1 we have right now.

L_1 = { ⟨M⟩ | M is a TM that accepts at least two strings }

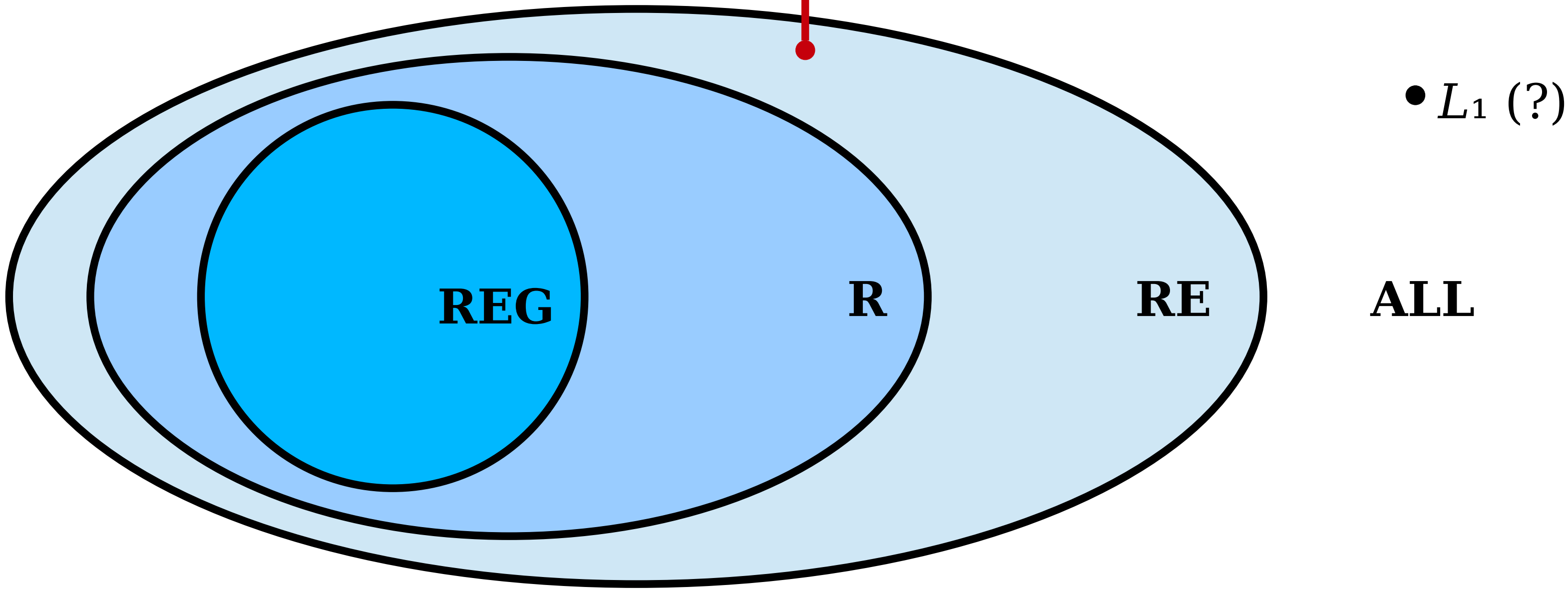
L_2 = { ⟨M⟩ | M is a TM that accepts exactly two strings }

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

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

RE: Languages with Verifiers

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



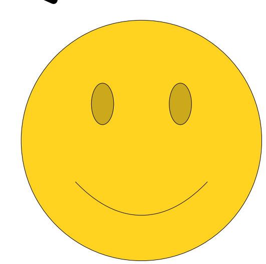
Imagine you have a string in L_1. What does that string look like?

L_1 = { ⟨M⟩ | M is a TM that accepts at least two strings }

L_2 = { ⟨M⟩ | M is a TM that accepts exactly two strings }

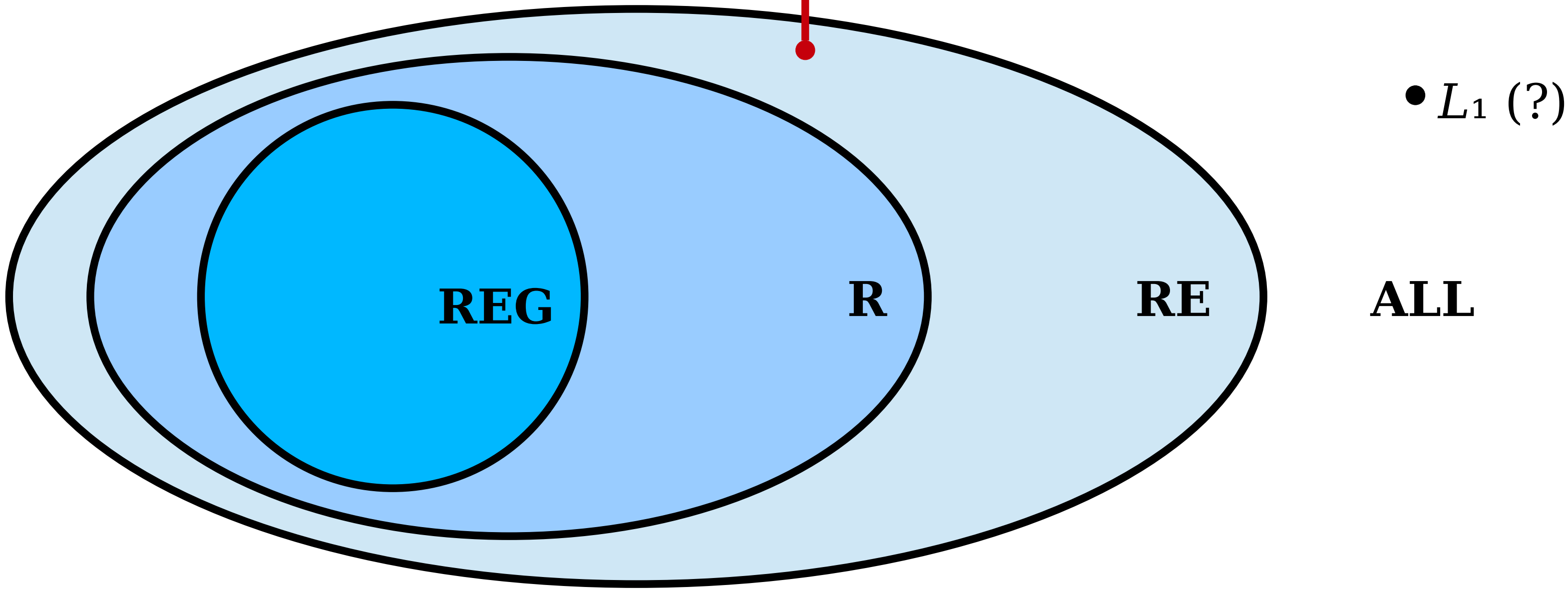
L_3 = { $\mathtt{a}^n\mathtt{b}^n$ | $n \in \mathbb{N}$ and $n > 1000$ }

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

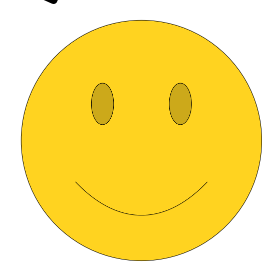


RE: Languages with Verifiers

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



Well, according to the definition of the language, any string in L_1 must encode a TM that accepts at least two strings.



$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

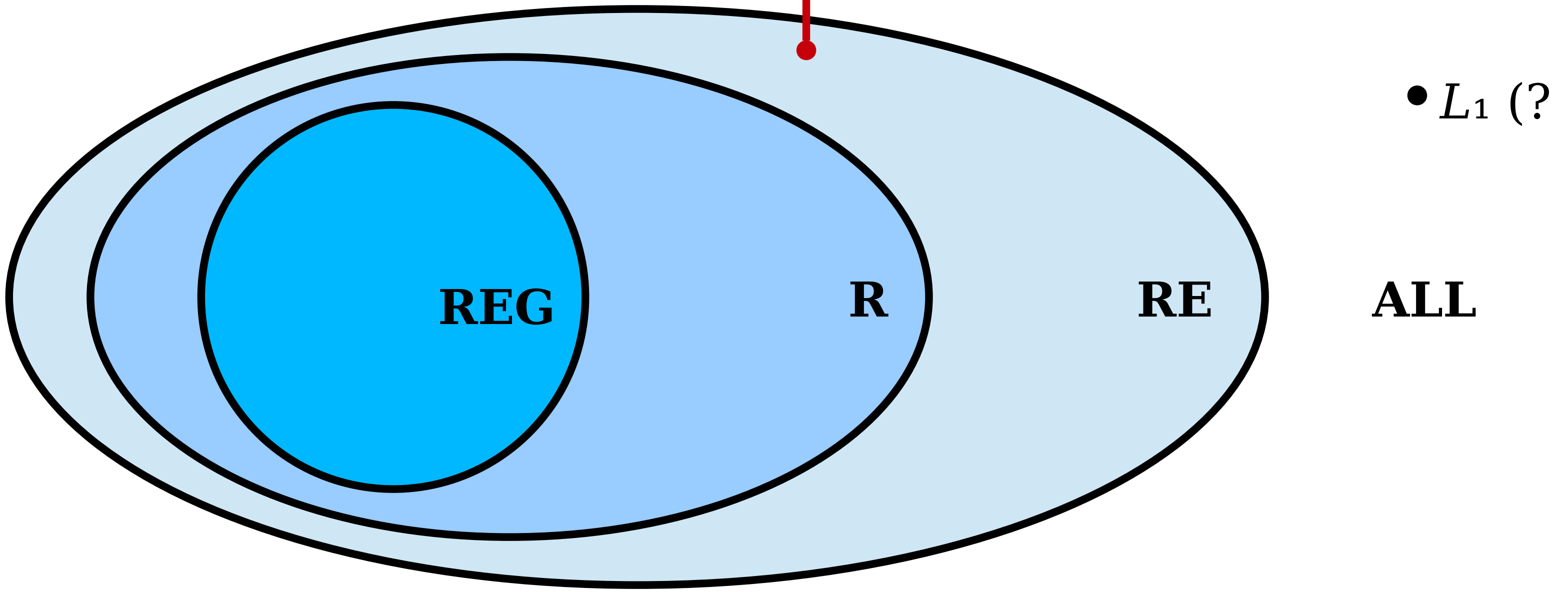
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

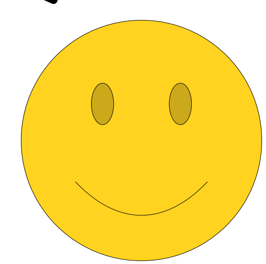
$L_4 = \{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N}$ and $n \leq 1000 \}$

RE: Languages with Verifiers

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



We can think of L_1 as "the language of TMs that accept at least two strings."



$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

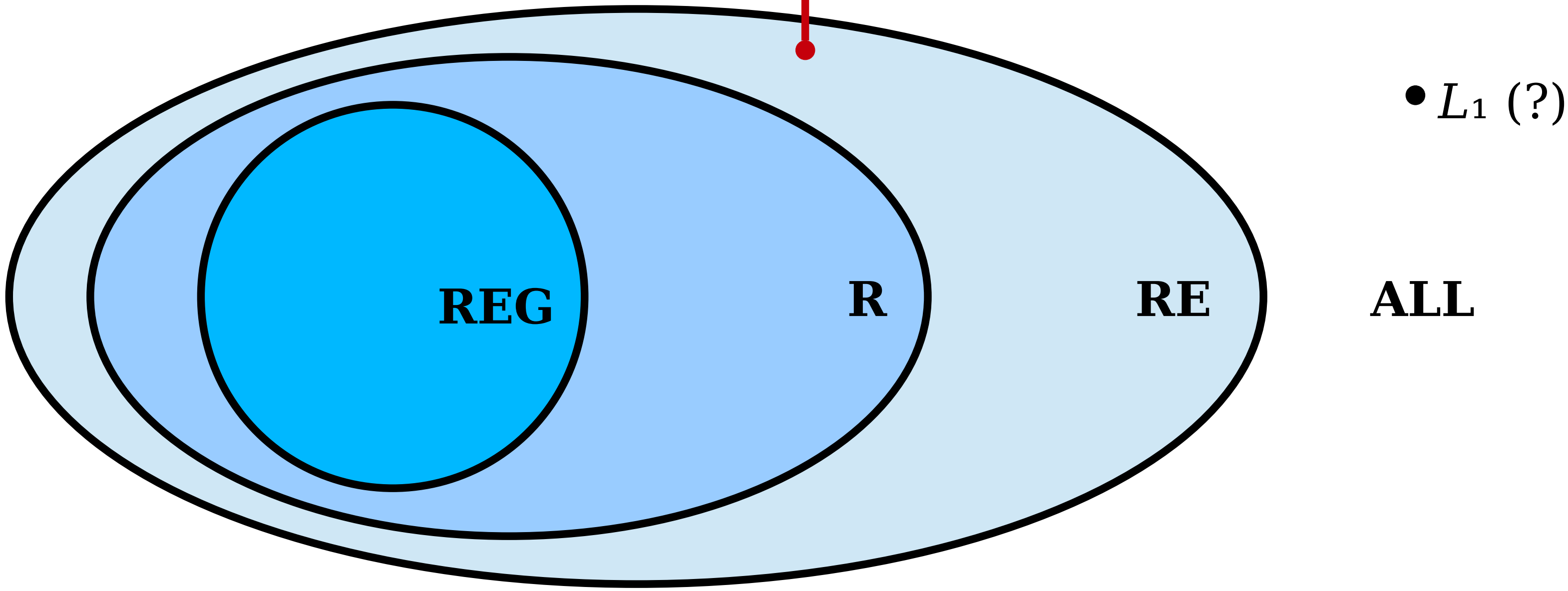
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

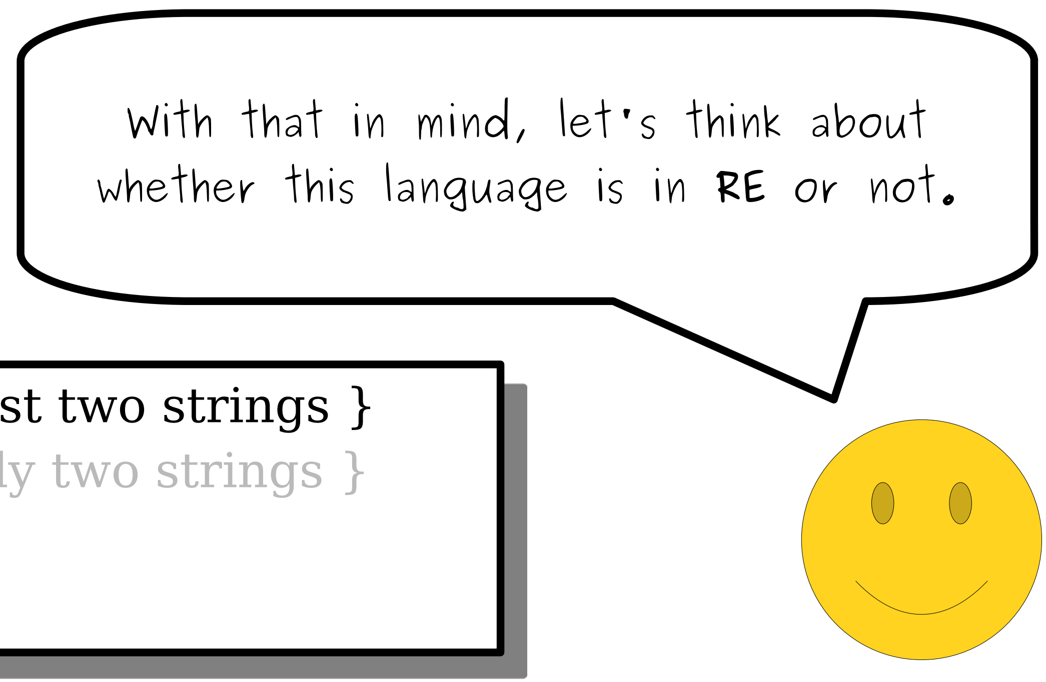
$L_3 = \{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N}$ and $n > 1000 \}$

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

RE: Languages with Verifiers

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





L_1 = { ⟨M⟩ | M is a TM that accepts at least two strings }

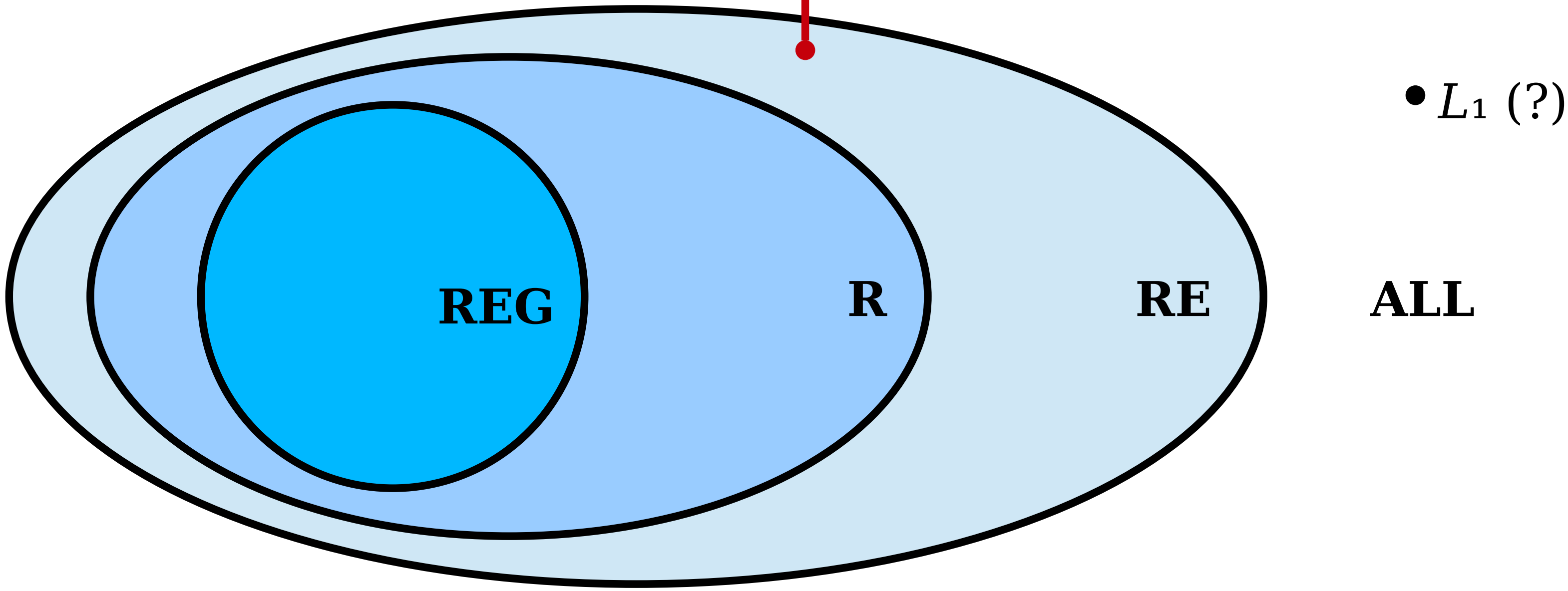
L_2 = { ⟨M⟩ | M is a TM that accepts exactly two strings }

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

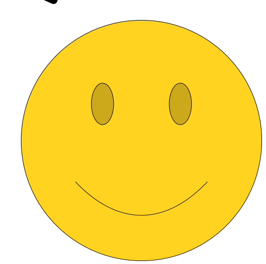
L_4 = { $\mathtt{a}^n\mathtt{b}^n$ | $n \in \mathbb{N}$ and $n \leq 1000$ }

RE: Languages with Verifiers

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



Let's imagine that we have a random TM and we are <u>*convinced*</u> that it accepts at least two strings.



$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

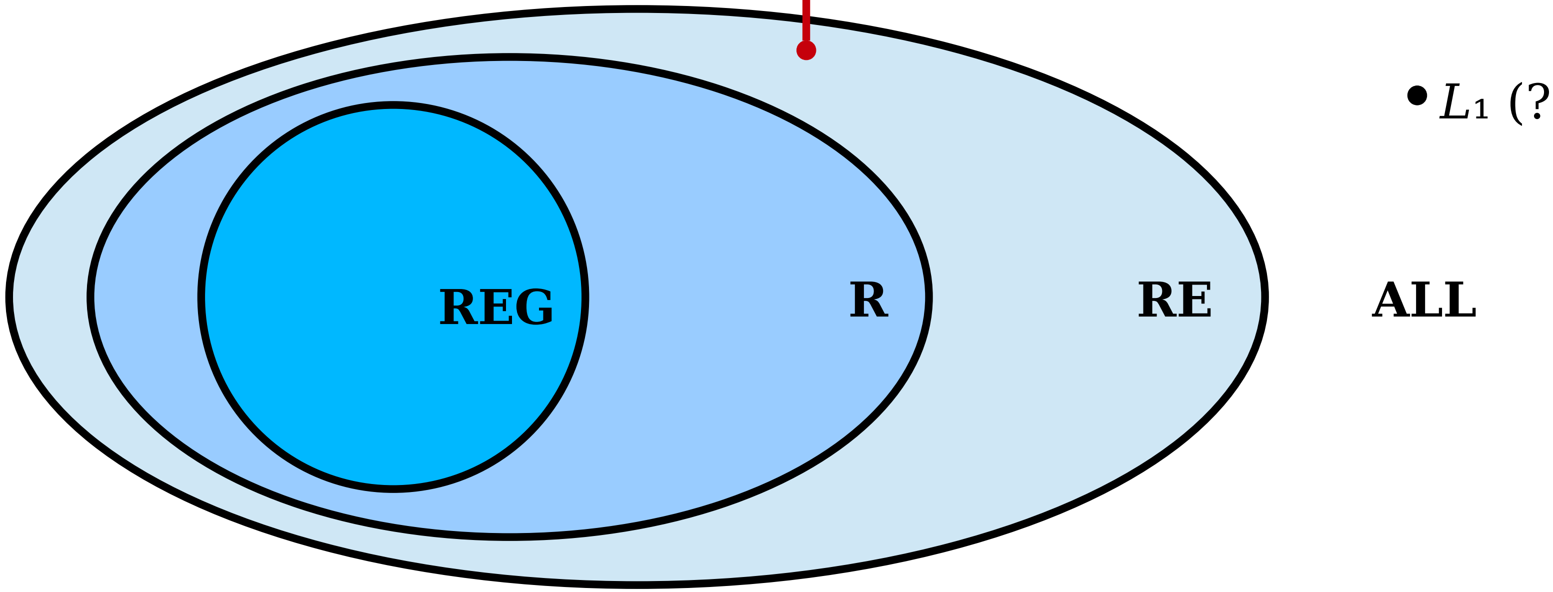
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

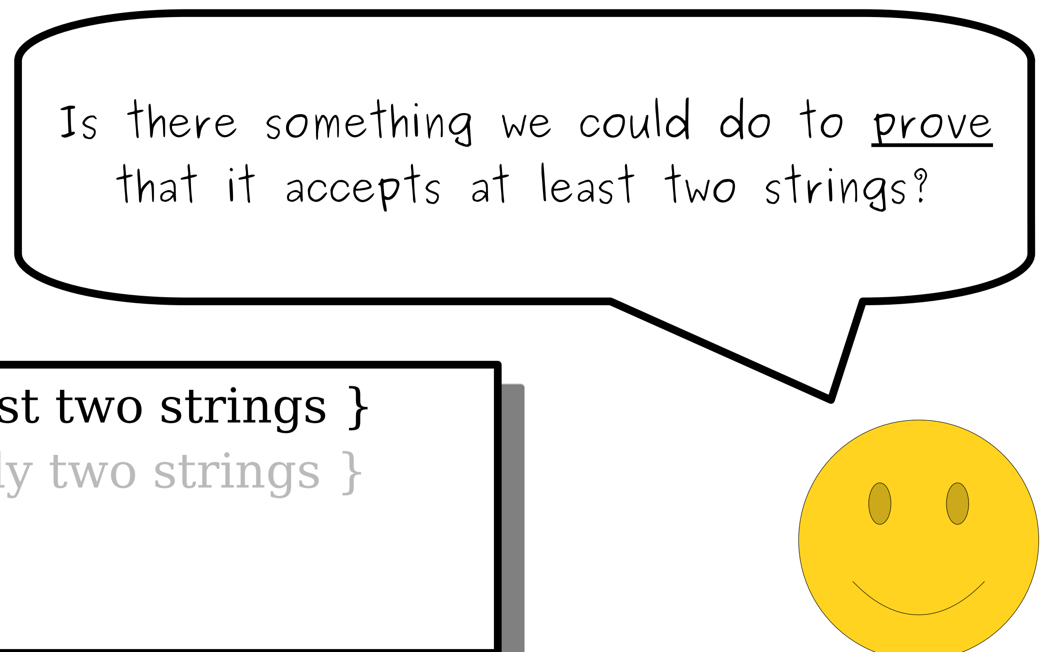
$L_3 = \{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N}$ and $n > 1000 \}$

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

RE: Languages with Verifiers

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





L_1 = { ⟨M⟩ | M is a TM that accepts at least two strings }

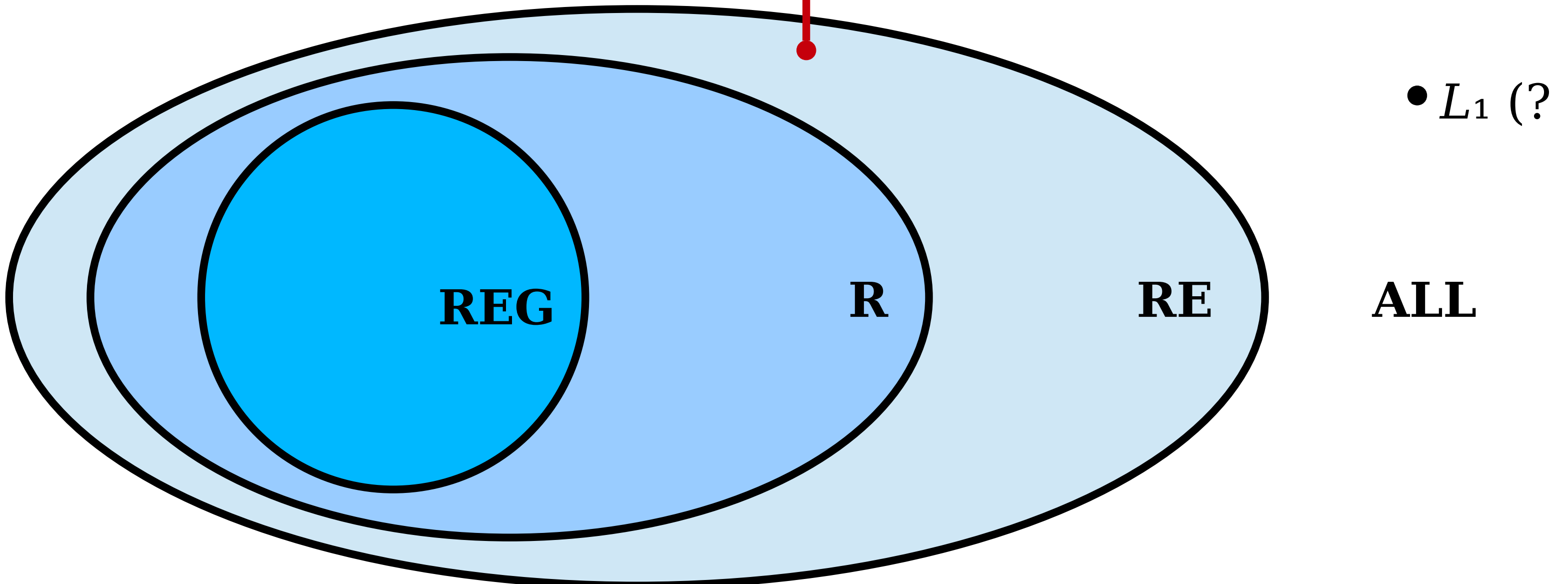
L_2 = { ⟨M⟩ | M is a TM that accepts exactly two strings }

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

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

RE: Languages with Verifiers

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



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?

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

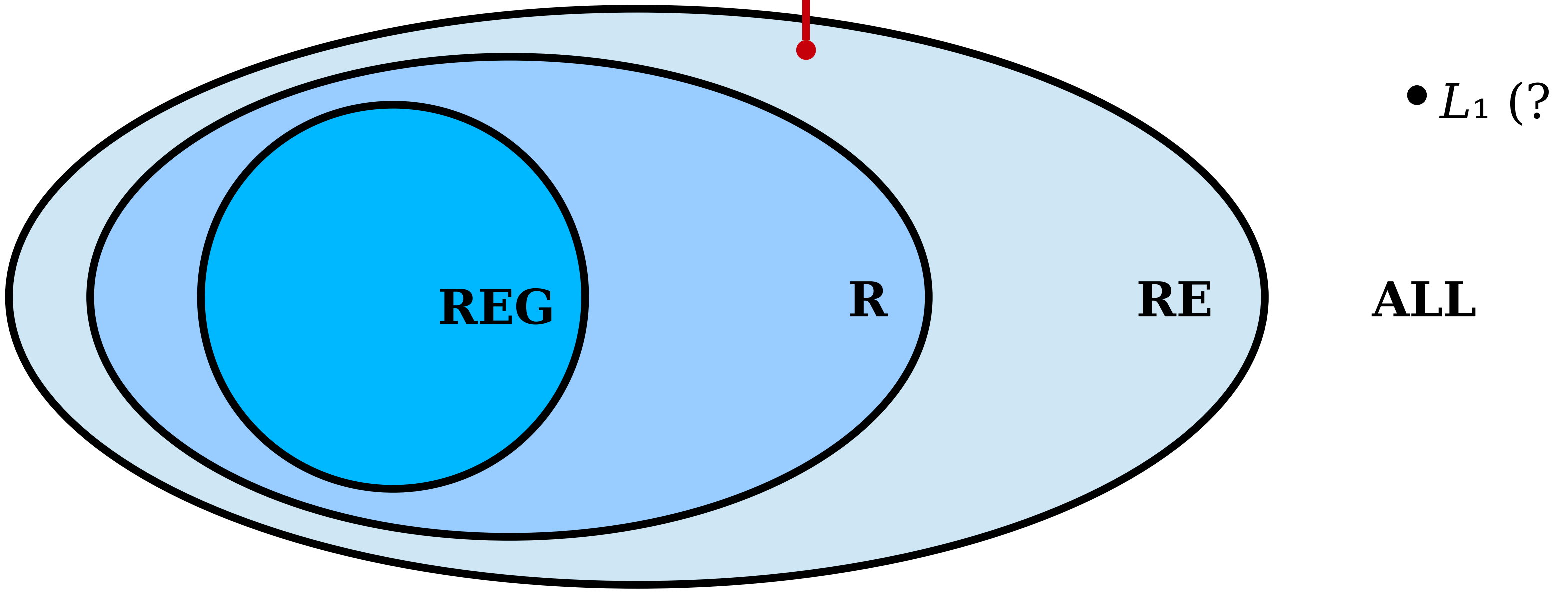
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

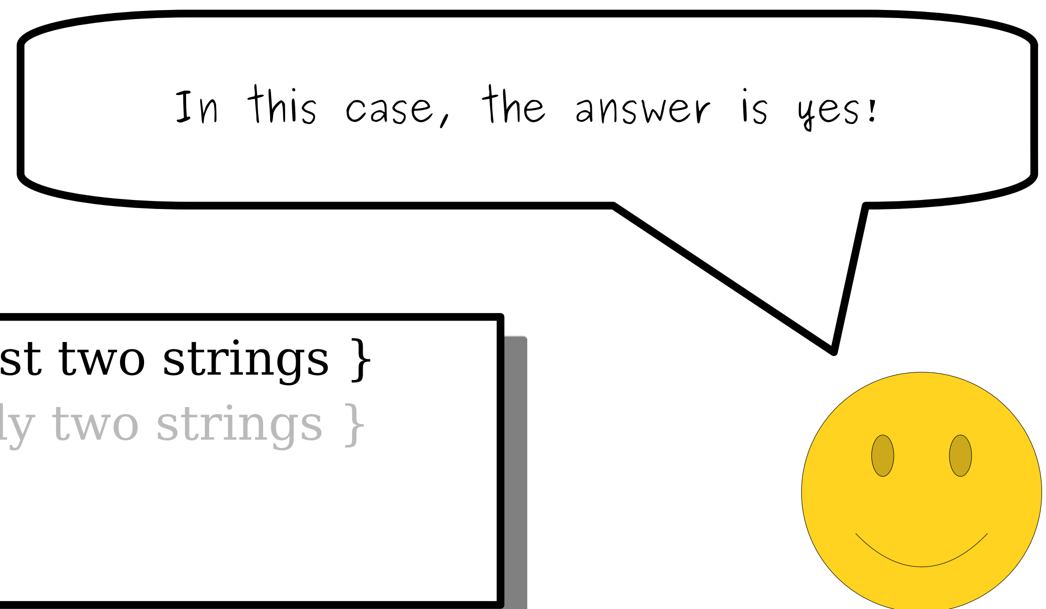
$L_3 = \{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N}$ and $n > 1000 \}$

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

RE: Languages with Verifiers

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





L_1 = { ⟨M⟩ | M is a TM that accepts at least two strings }

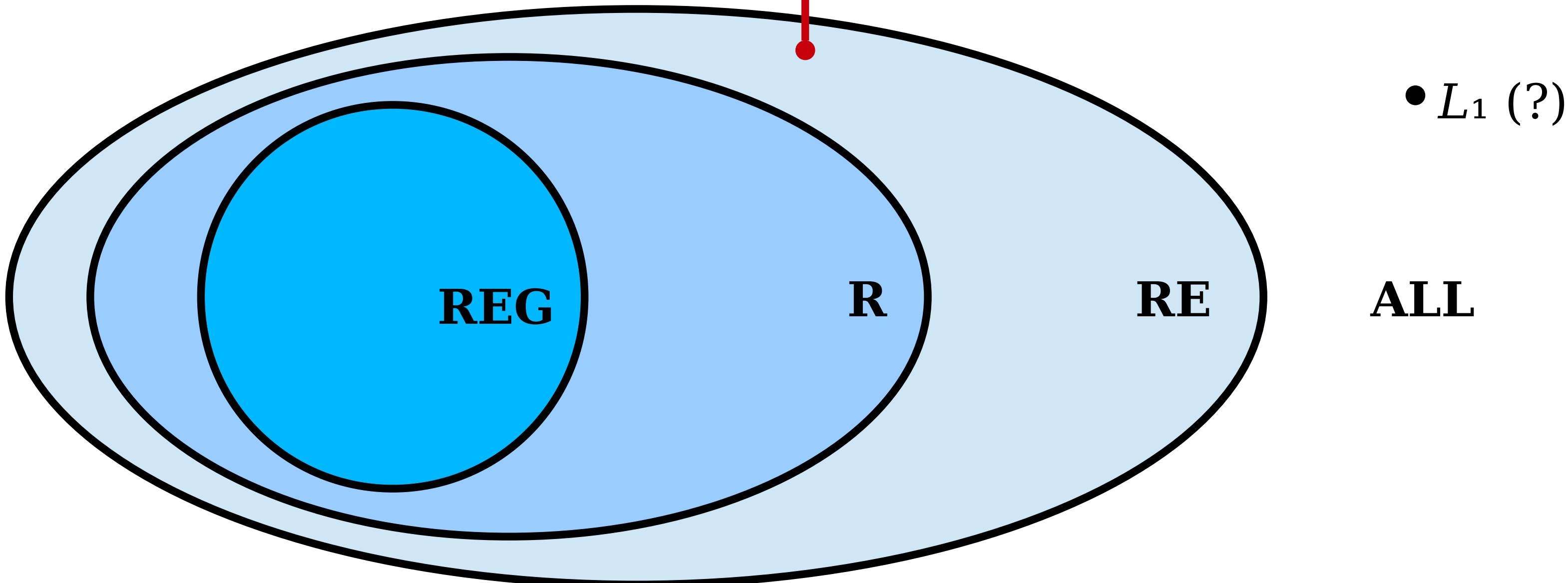
L_2 = { ⟨M⟩ | M is a TM that accepts exactly two strings }

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

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

RE: Languages with Verifiers

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



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.

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

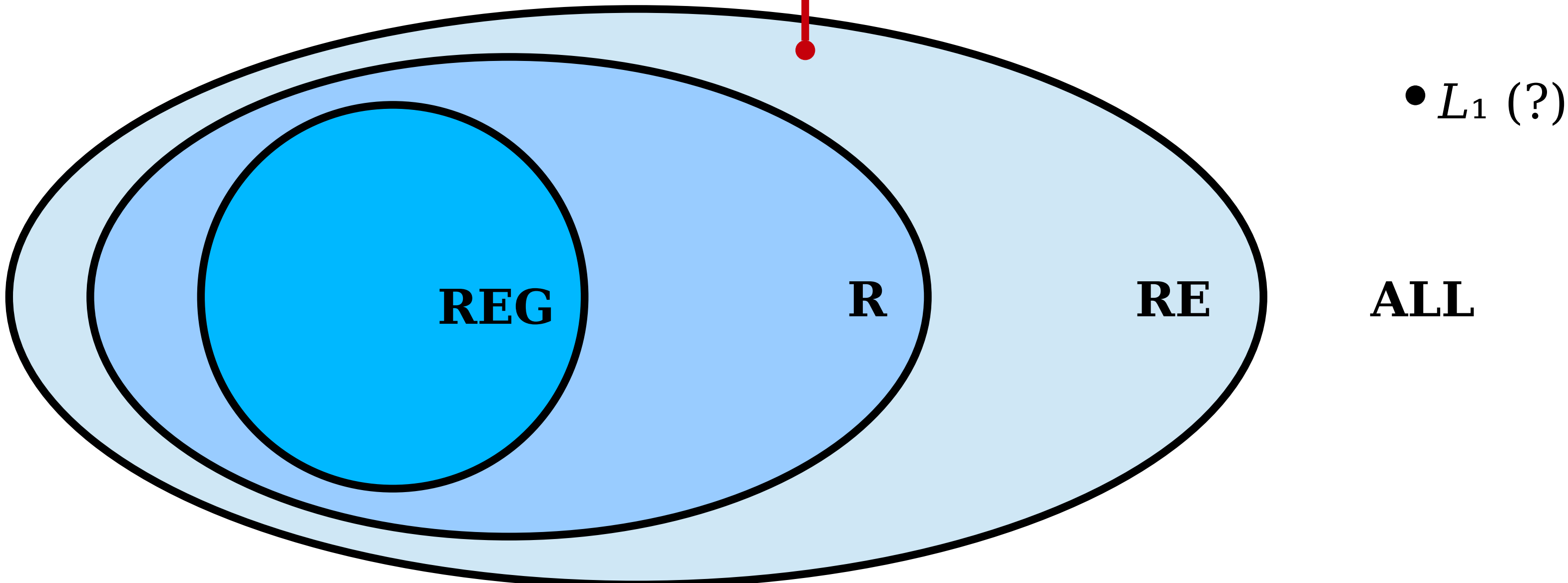
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

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

RE: Languages with Verifiers

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



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!

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

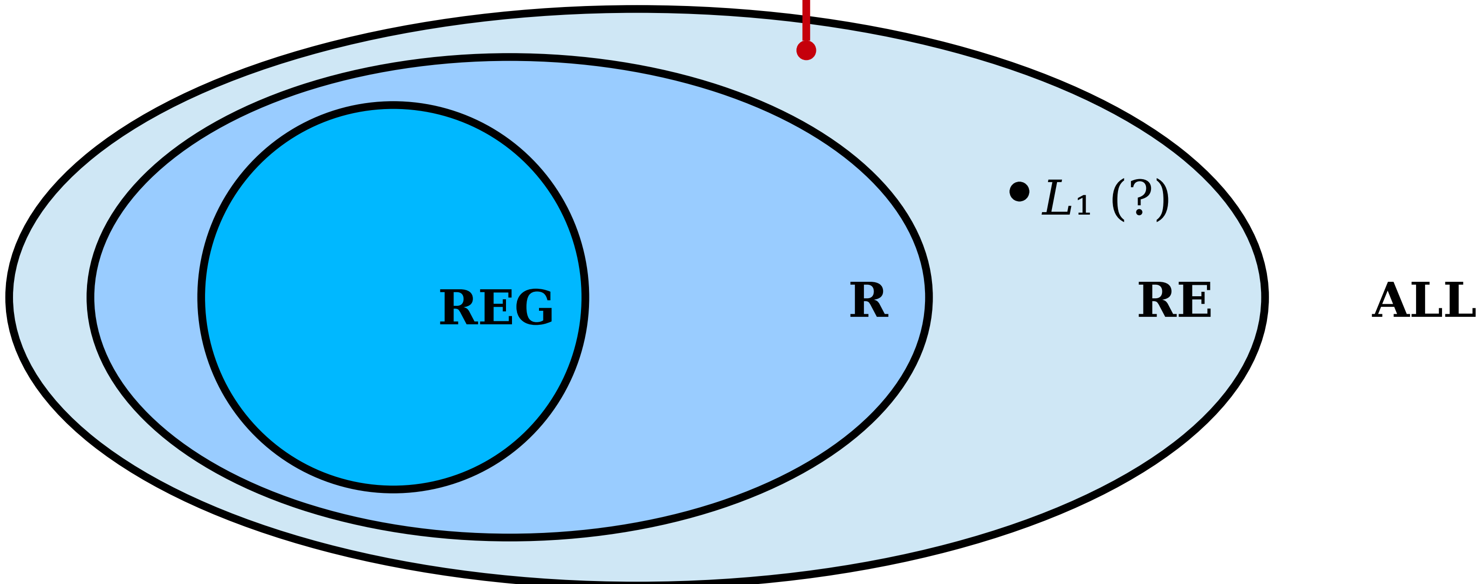
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

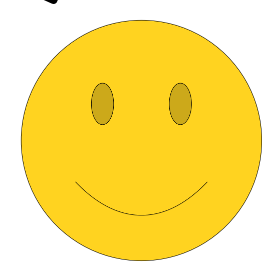
$L_4 = \{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N}$ and $n \leq 1000 \}$

RE: Languages with Verifiers

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



So, going off this intuition, we can be reasonably confident that the language L_1 is indeed in **RE**.



L_1 = { ⟨M⟩ | M is a TM that accepts at least two strings }

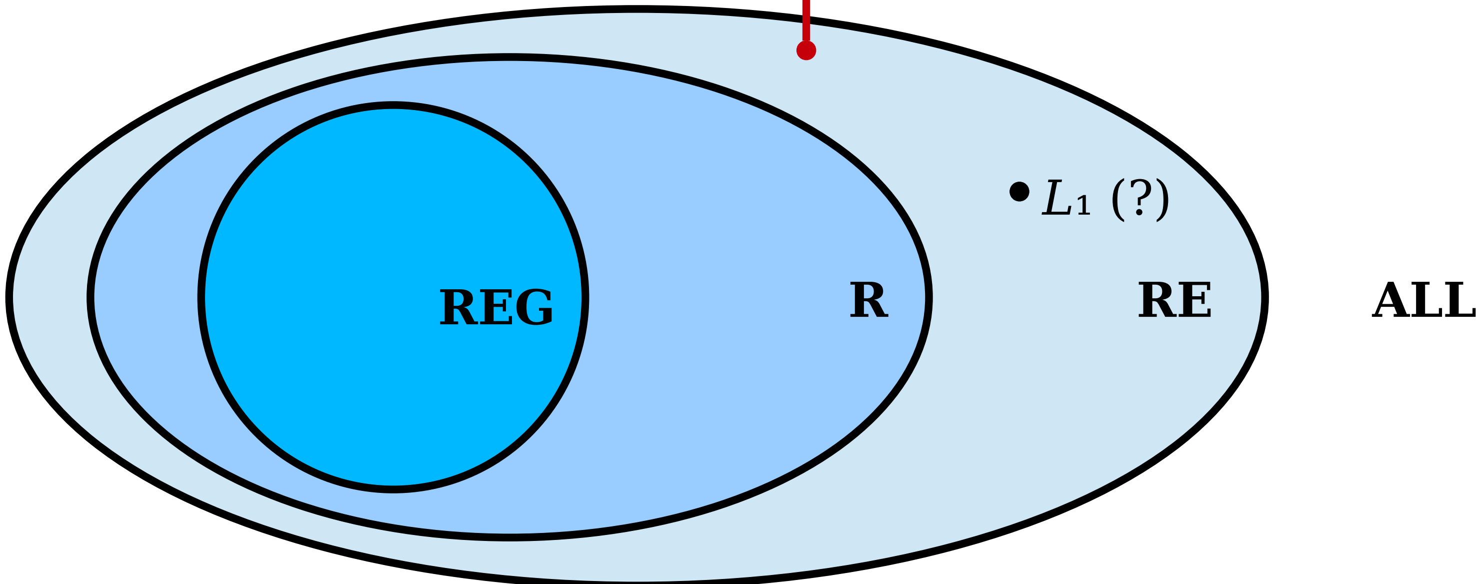
L_2 = { ⟨M⟩ | M is a TM that accepts exactly two strings }

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

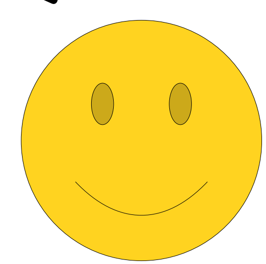
$L_4 = \{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$

RE: Languages with Verifiers

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



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 **RE**.



$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

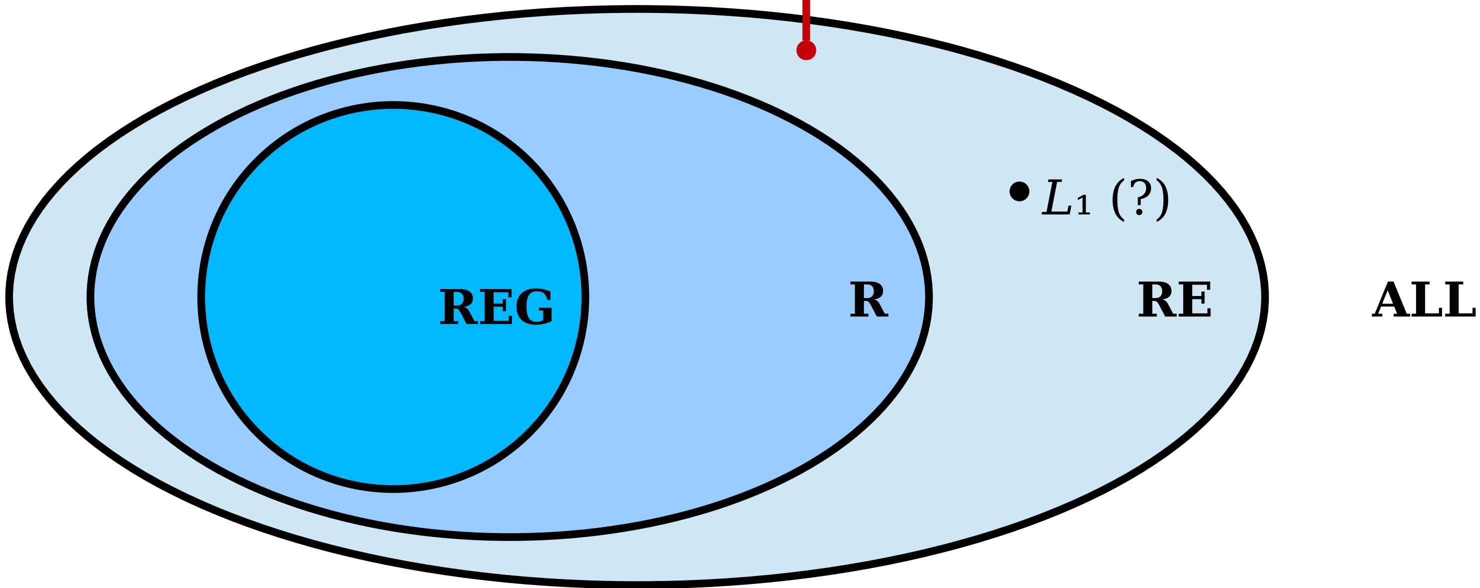
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

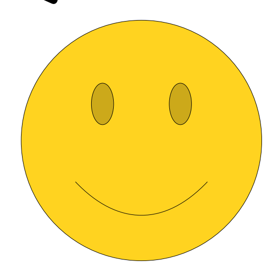
$L_4 = \{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N}$ and $n \leq 1000 \}$

RE: Languages with Verifiers

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



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.



$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

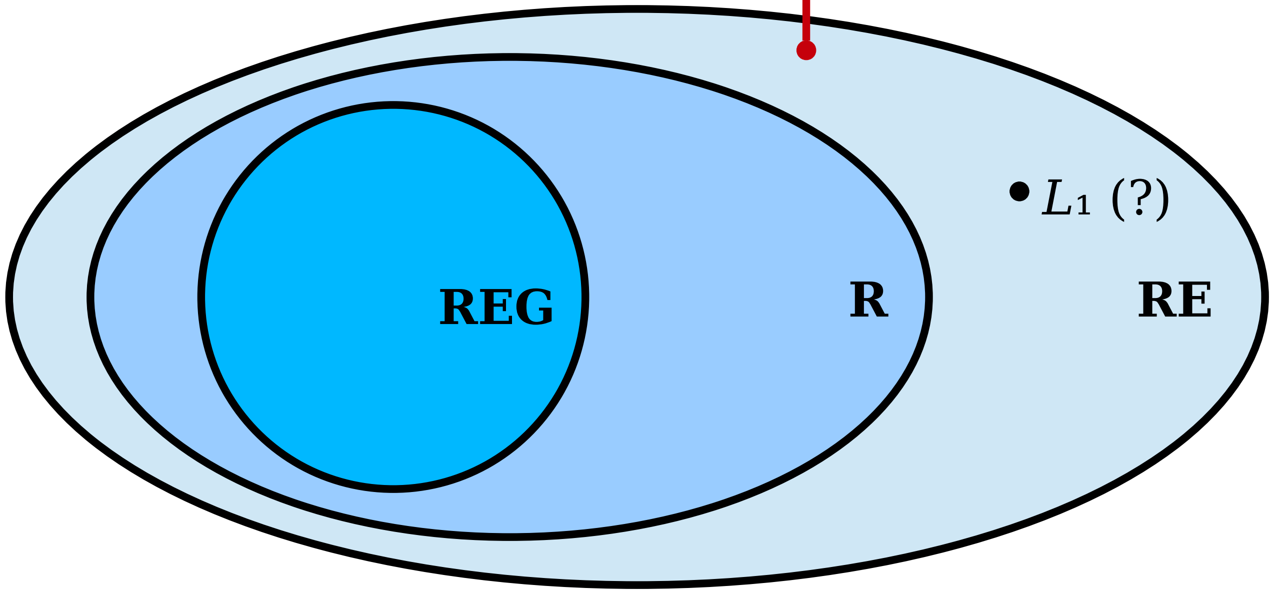
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

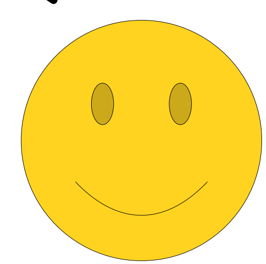
$L_4 = \{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N}$ and $n \leq 1000 \}$

RE: Languages with Verifiers

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



The idea would go something like this.



$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

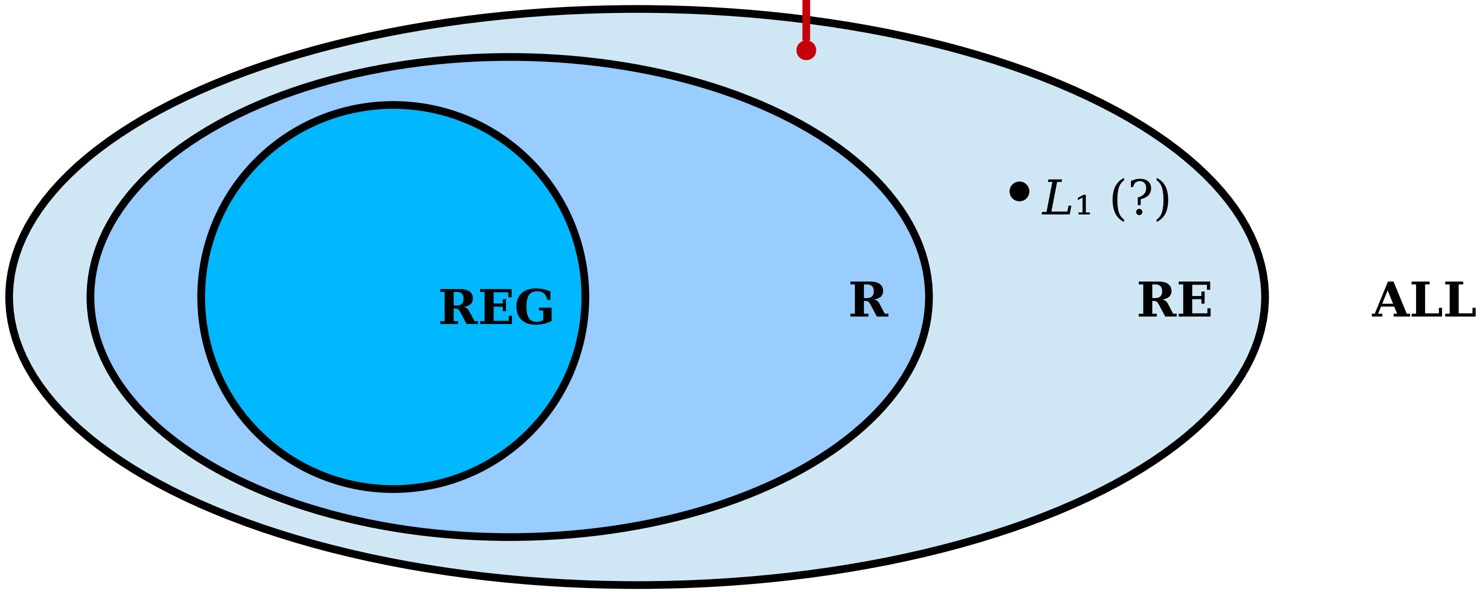
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

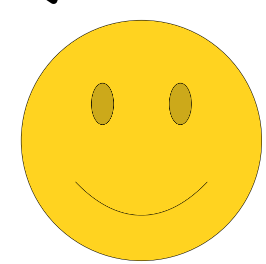
$L_4 = \{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N}$ and $n \leq 1000 \}$

RE: Languages with Verifiers

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



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 = { ⟨M⟩ | M is a TM that accepts at least two strings }

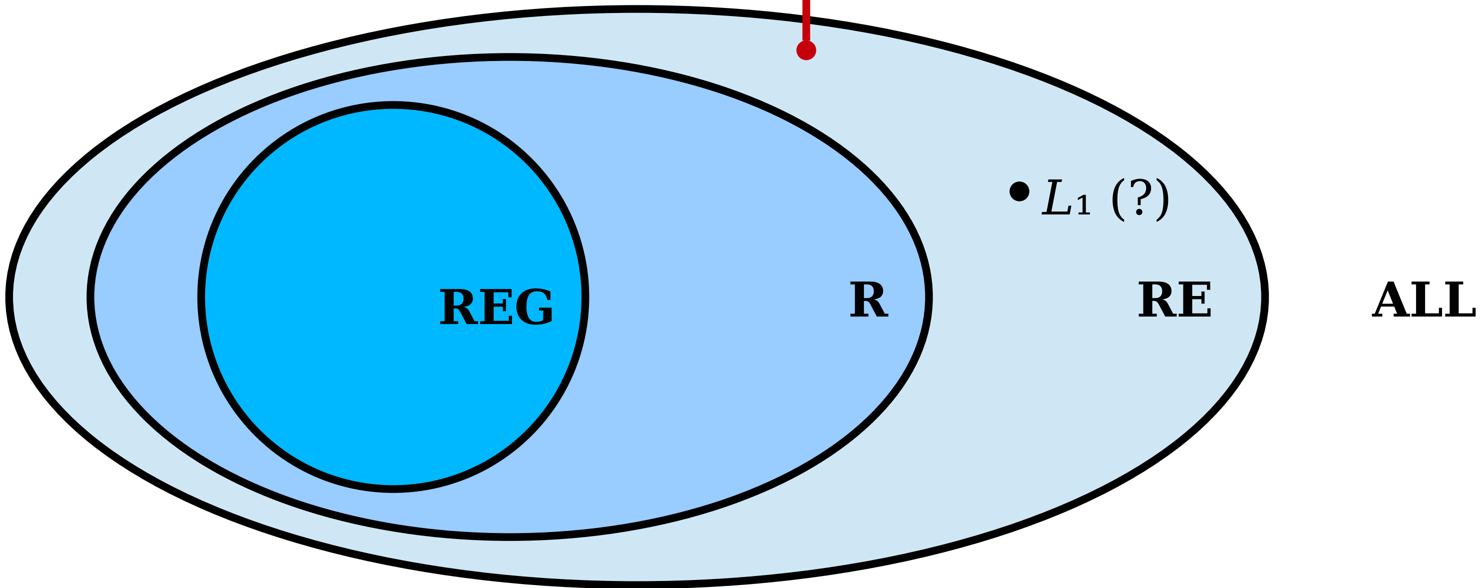
L_2 = { ⟨M⟩ | M is a TM that accepts exactly two strings }

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

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

RE: Languages with Verifiers

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



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.

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

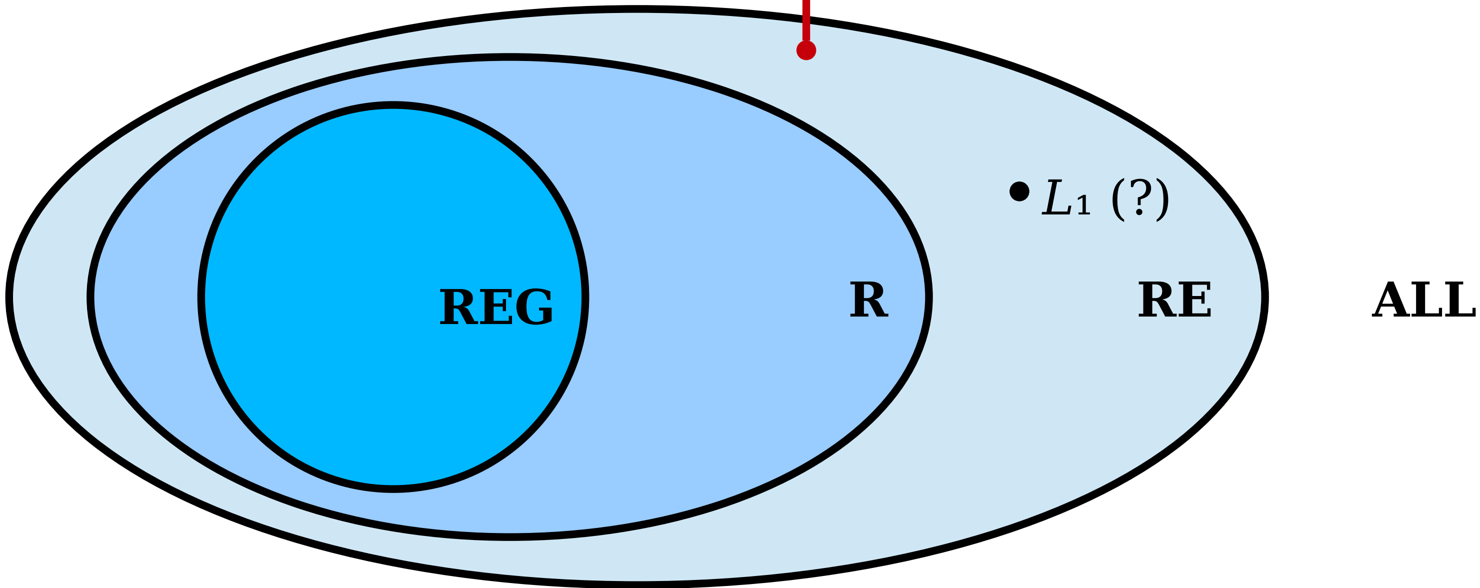
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

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

RE: Languages with Verifiers

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

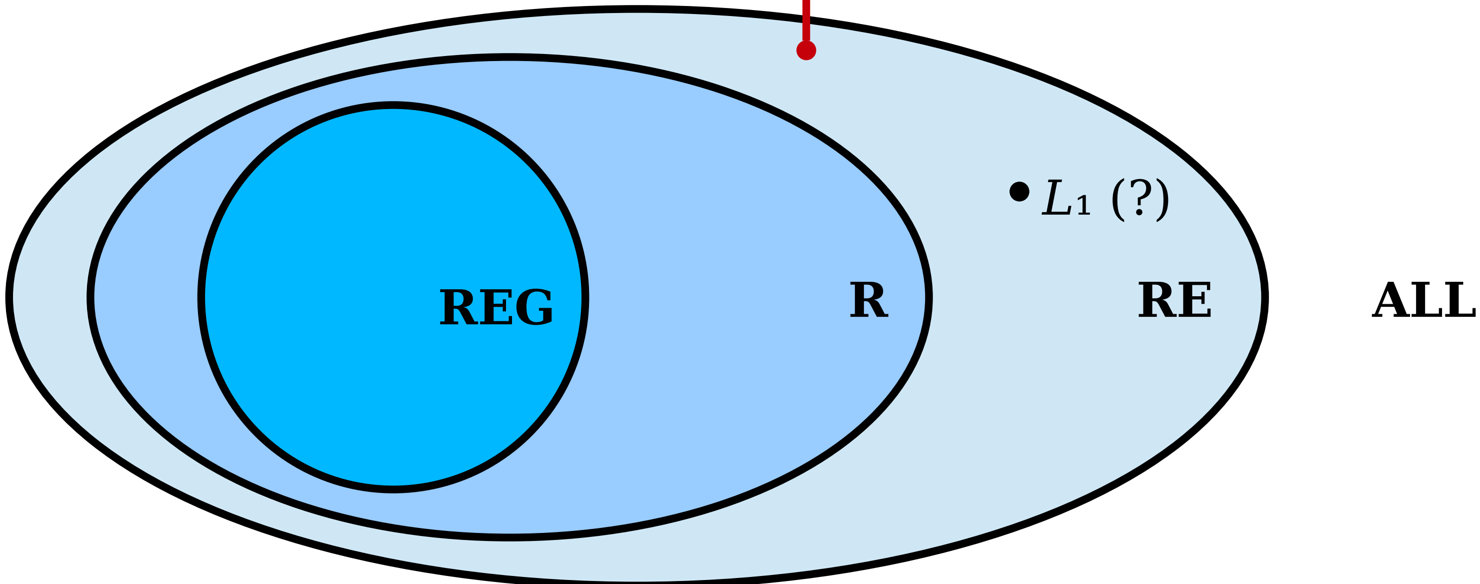


So the verifier would take in as input the TM M, two strings w_1 and w_2, and a number of steps n, and could run M on the stings w_1 and w_2 for up to n steps.

L_1 = { ⟨M⟩ | M is a TM that accepts at least two strings }
L_2 = { ⟨M⟩ | M is a TM that accepts exactly two strings }
L_3 = { $\mathbf{a}^n\mathbf{b}^n$ | $n \in \mathbb{N}$ and $n > 1000$ }
L_4 = { $\mathbf{a}^n\mathbf{b}^n$ | $n \in \mathbb{N}$ and $n \leq 1000$ }

RE: Languages with Verifiers

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



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.

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

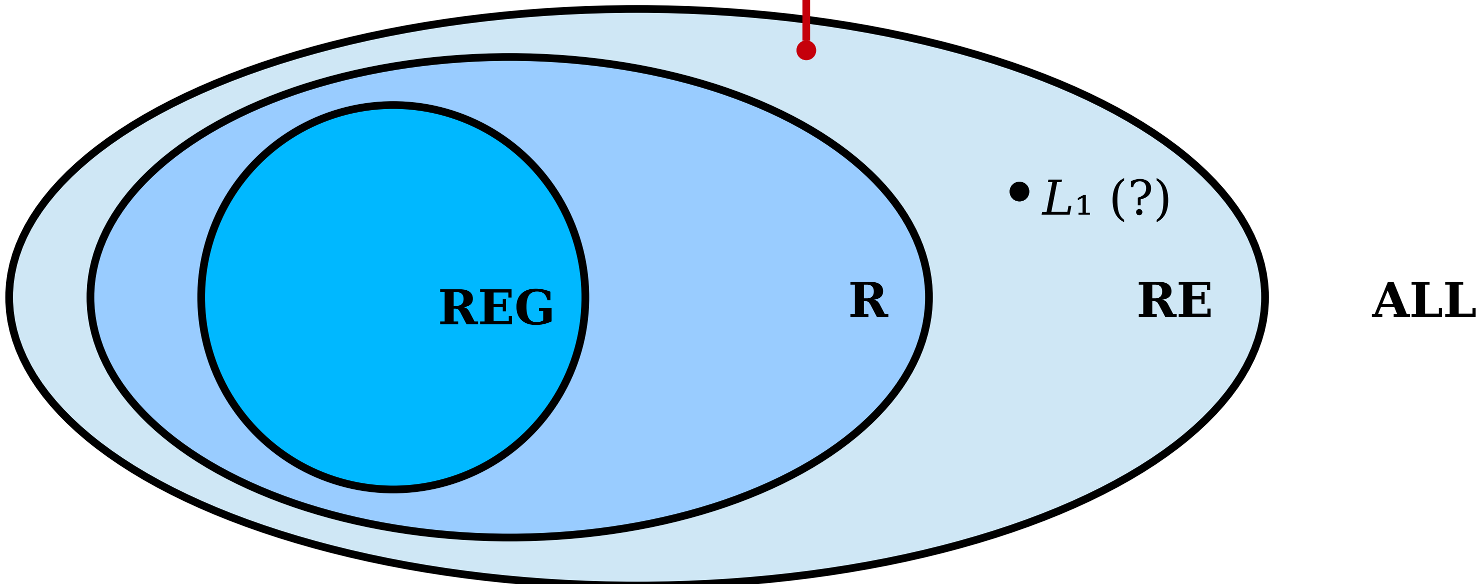
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

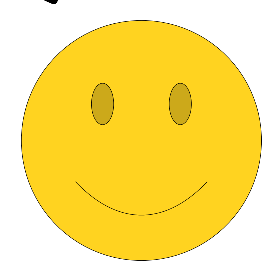
$L_4 = \{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N}$ and $n \leq 1000 \}$

RE: Languages with Verifiers

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



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.



$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

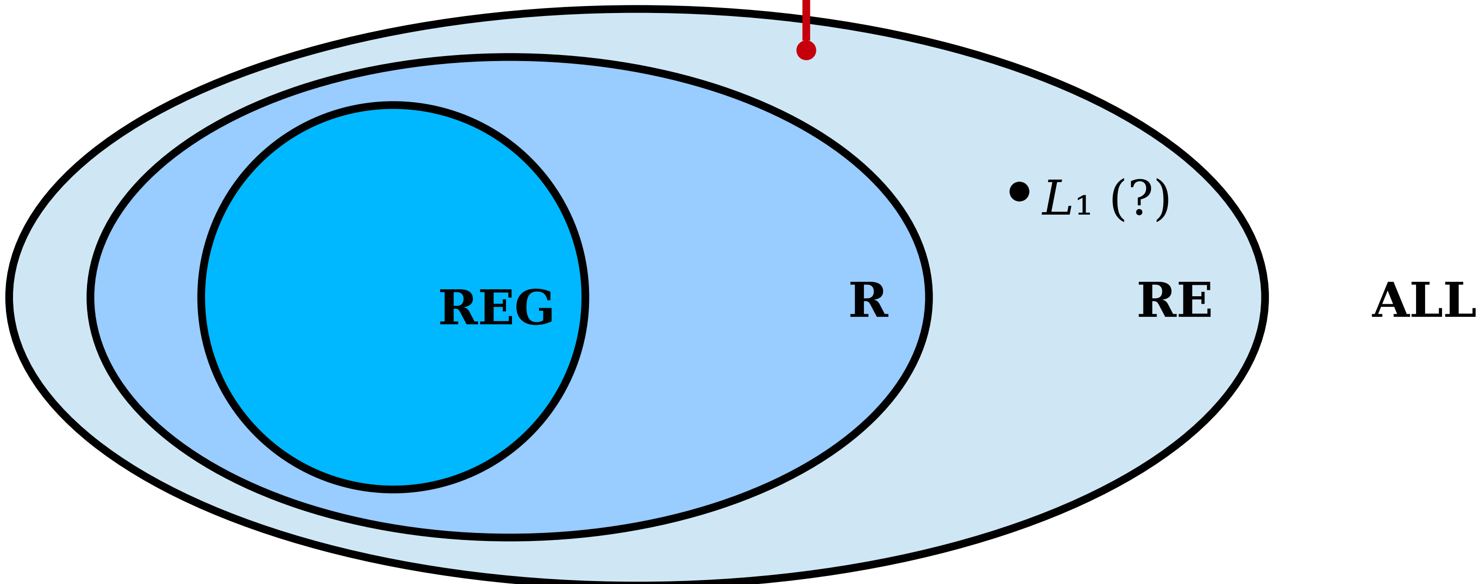
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

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

RE: Languages with Verifiers

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



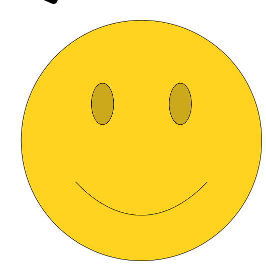
If you wanted to write this up as a formal proof, it's a good exercise! For now, though, we're just going to continue working through figuring out where this language goes on the Lava Diagram.

L_1 = { $\langle M \rangle$ | M is a TM that accepts at least two strings }

L_2 = { $\langle M \rangle$ | M is a TM that accepts exactly two strings }

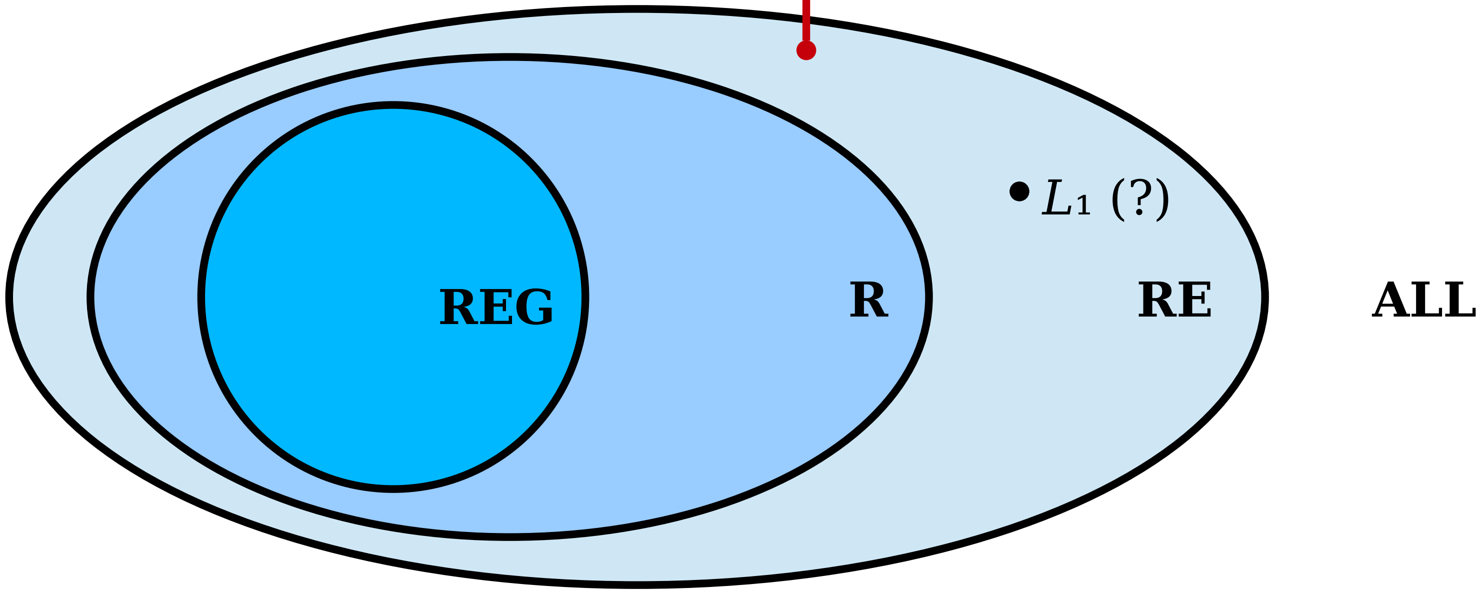
L_3 = { $\mathbf{a}^n\mathbf{b}^n$ | $n \in \mathbb{N}$ and $n > 1000$ }

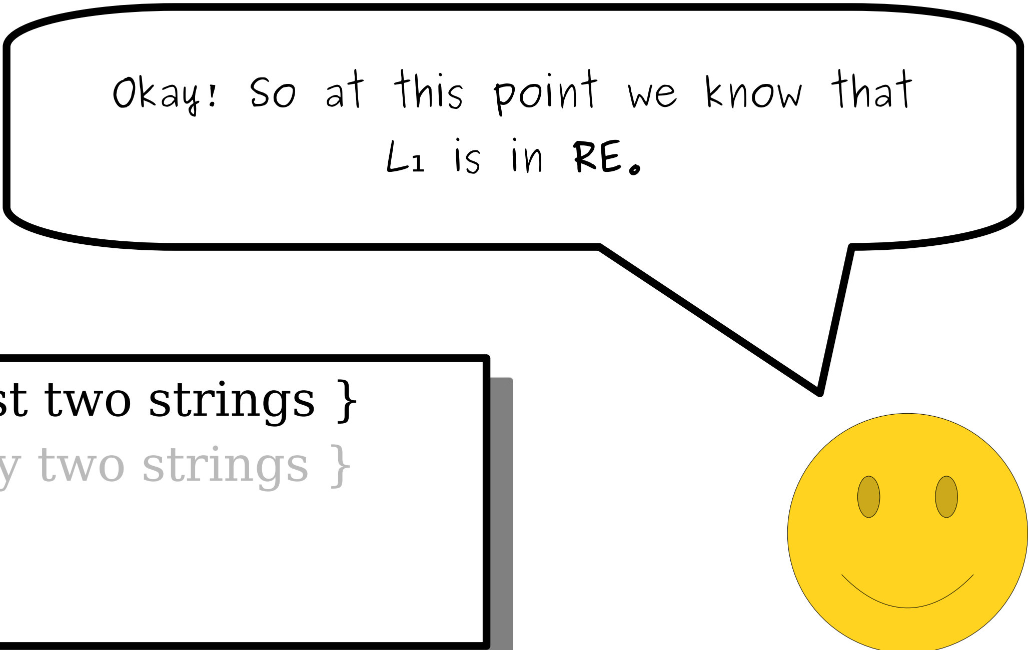
L_4 = { $\mathbf{a}^n\mathbf{b}^n$ | $n \in \mathbb{N}$ and $n \leq 1000$ }



RE: Languages with Verifiers

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





L_1 = { ⟨M⟩ | M is a TM that accepts at least two strings }

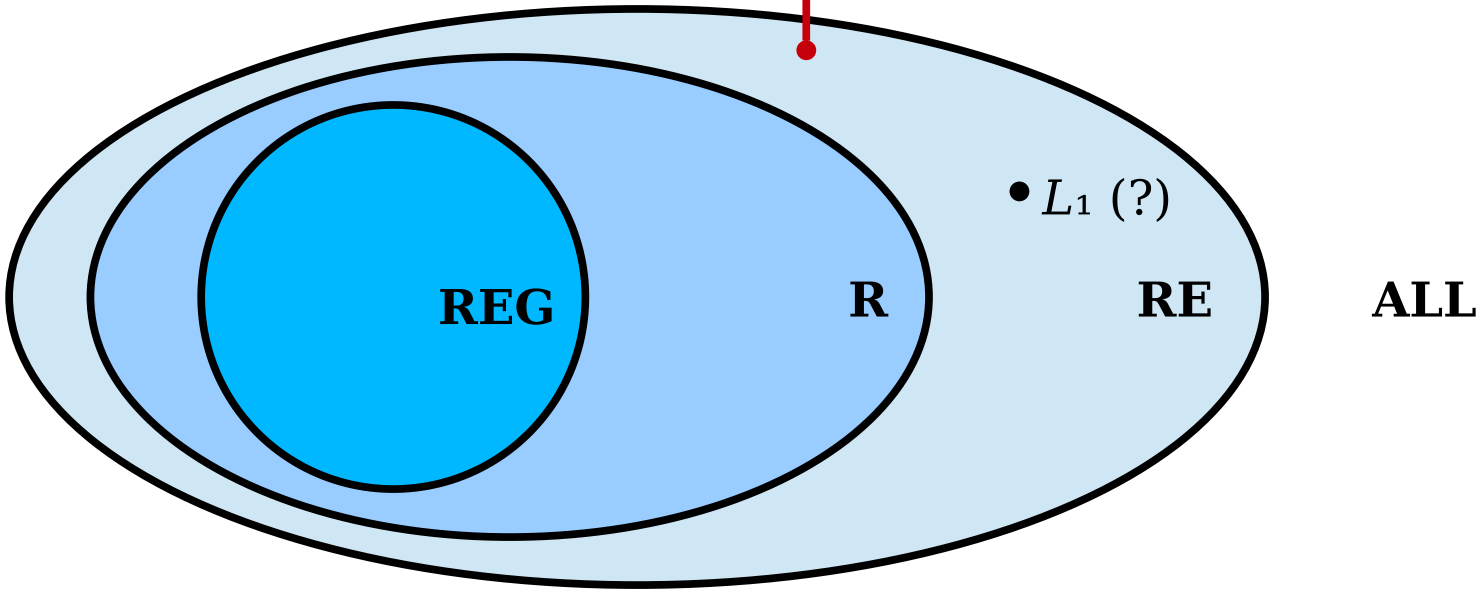
L_2 = { ⟨M⟩ | M is a TM that accepts exactly two strings }

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

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

RE: Languages with Verifiers

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



The next step is to determine whether it's also in class **R**.

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

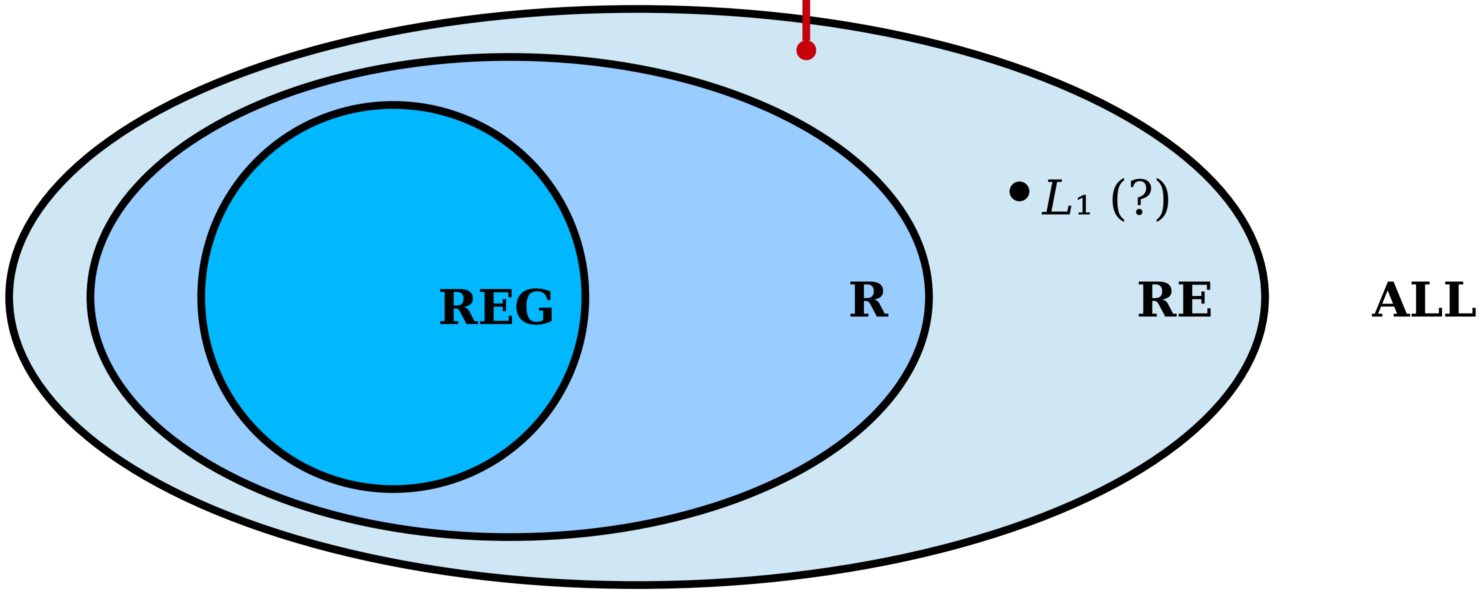
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

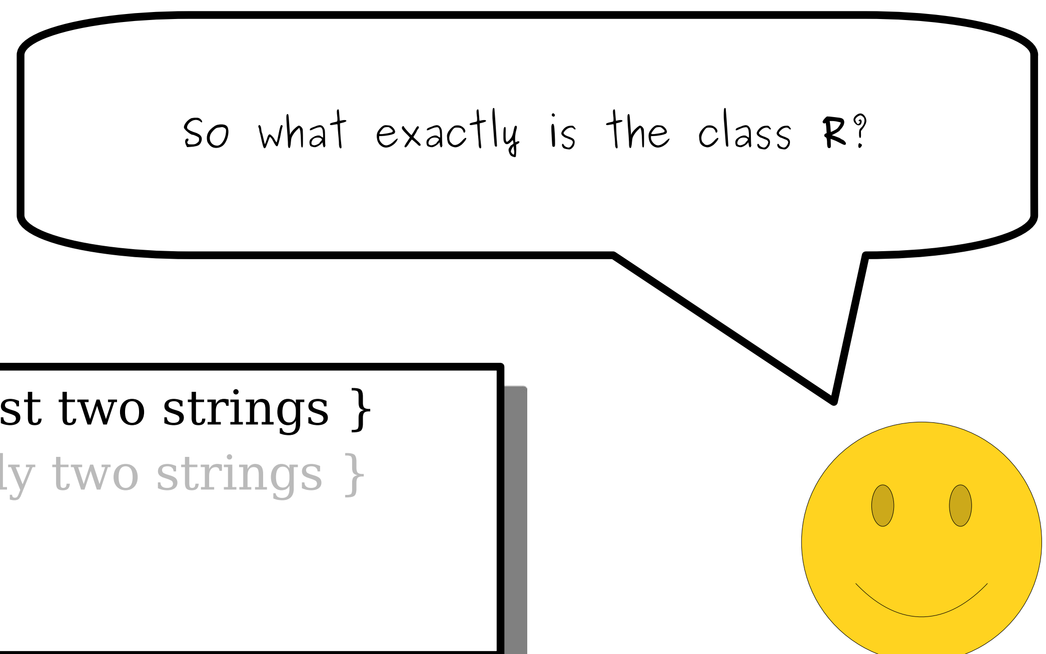
$L_3 = \{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N}$ and $n > 1000 \}$

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

RE: Languages with Verifiers

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





L_1 = { ⟨M⟩ | M is a TM that accepts at least two strings }

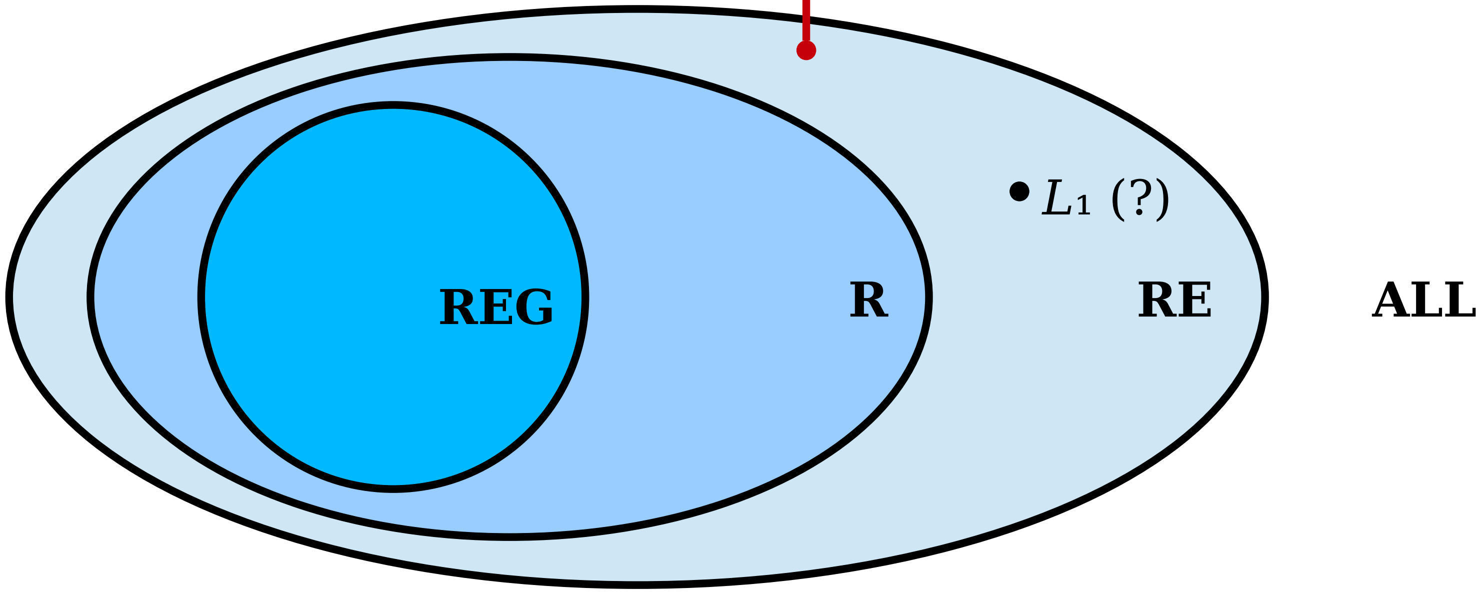
L_2 = { ⟨M⟩ | M is a TM that accepts exactly two strings }

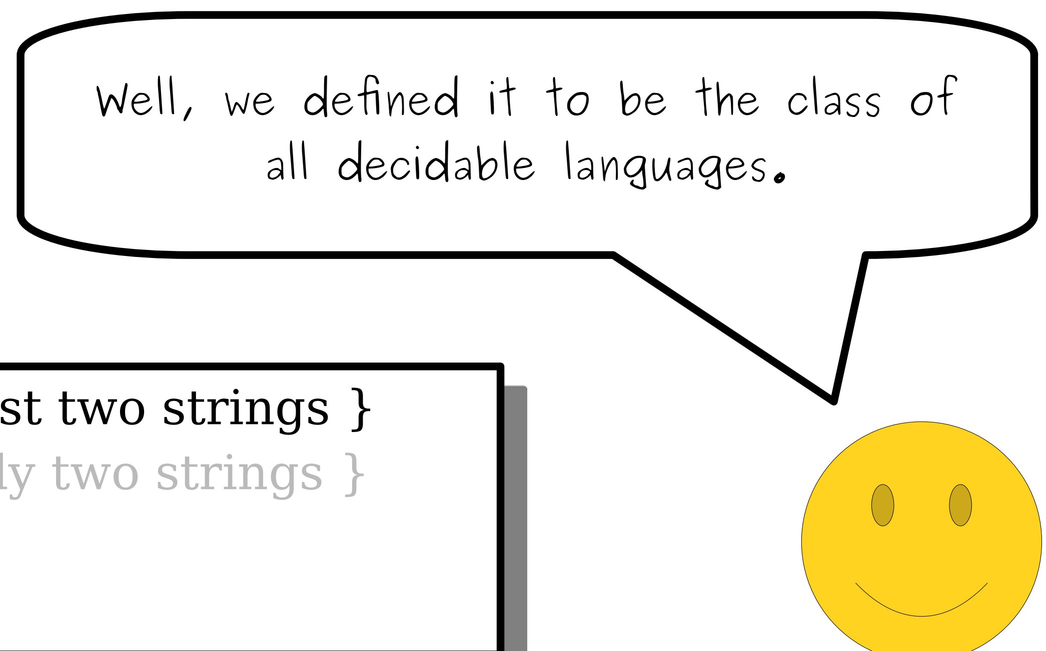
L_3 = { $\mathbf{a}^n\mathbf{b}^n$ | $n \in \mathbb{N}$ and $n > 1000$ }

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

RE: Languages with Verifiers

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





L_1 = { $\langle M \rangle$ | M is a TM that accepts at least two strings }

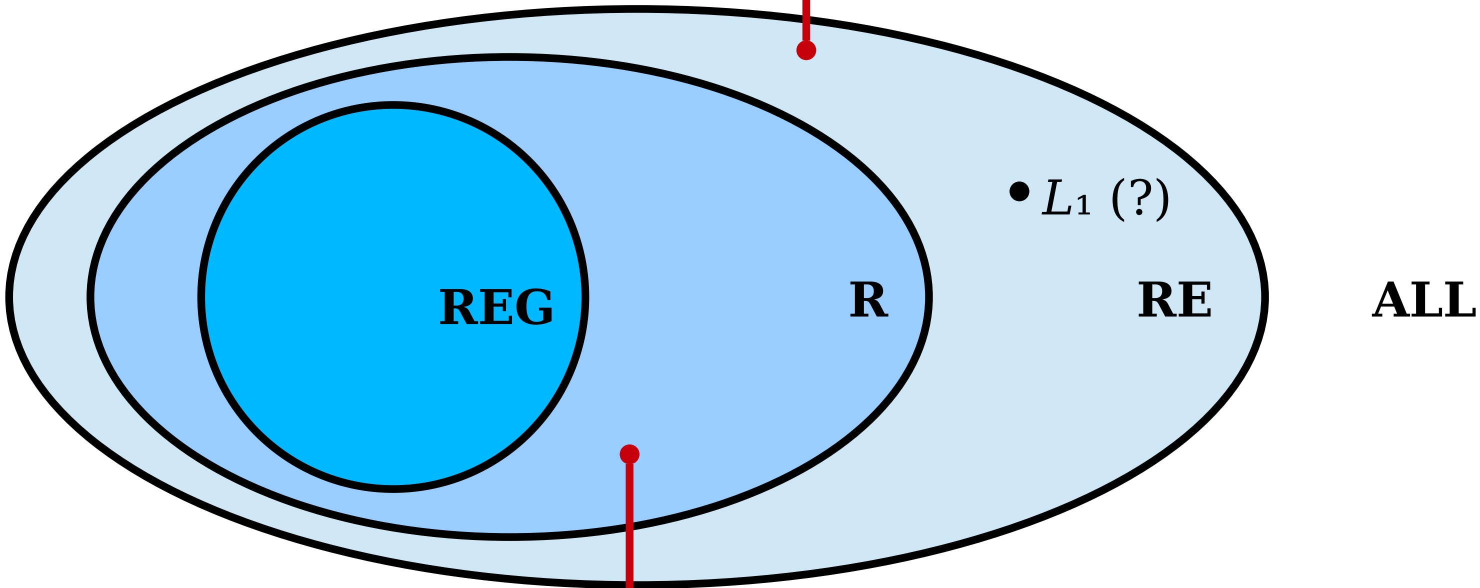
L_2 = { $\langle M \rangle$ | M is a TM that accepts exactly two strings }

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

L_4 = { $\mathbf{a}^n\mathbf{b}^n$ | $n \in \mathbb{N}$ 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

That means that it's the class of all languages that have deciders.

L_1 = { ⟨M⟩ | M is a TM that accepts at least two strings }

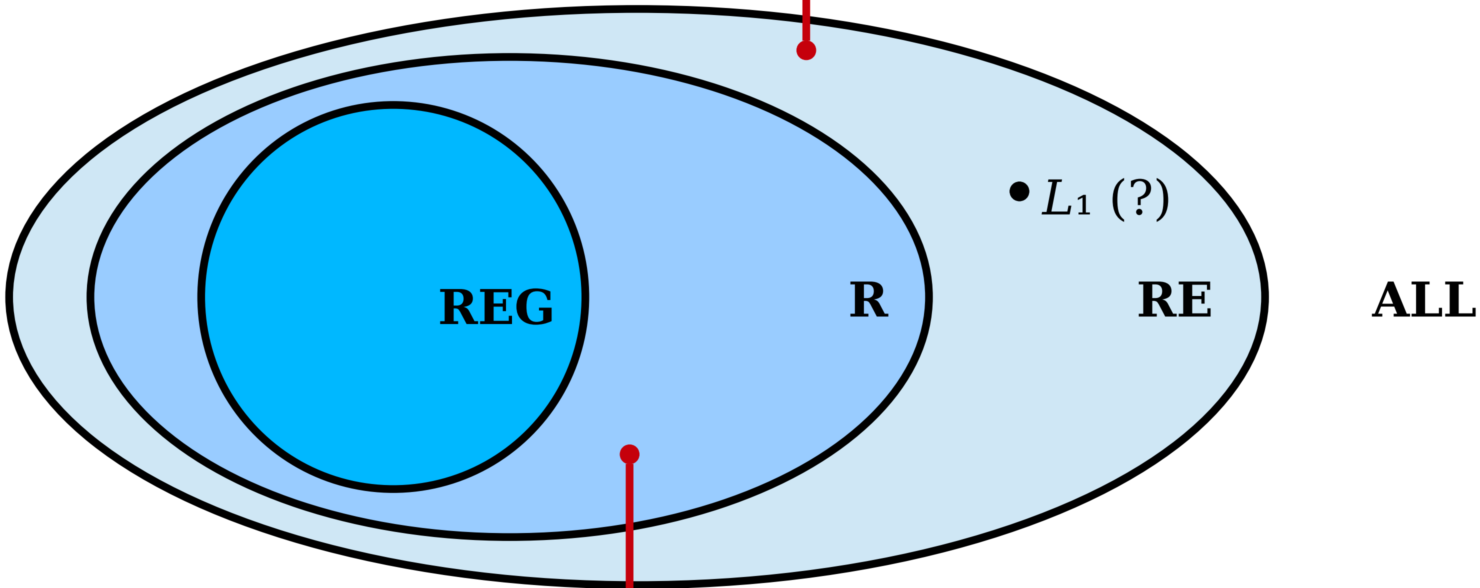
L_2 = { ⟨M⟩ | M is a TM that accepts exactly two strings }

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

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

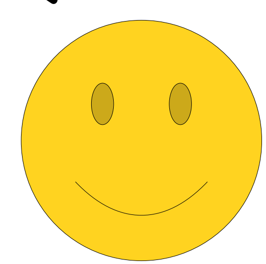
RE: Languages with Verifiers

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



R: Languages with Deciders

You can reason about whether a language belongs to class R by thinking about whether you could build a decider for it.



$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

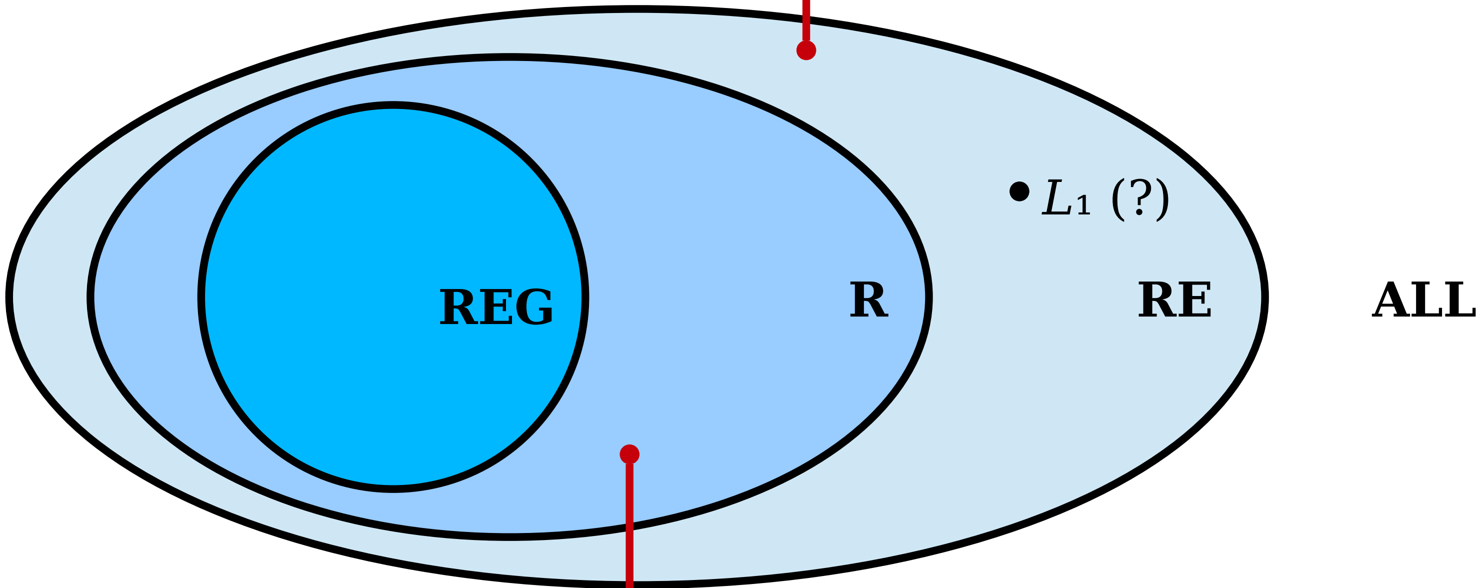
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

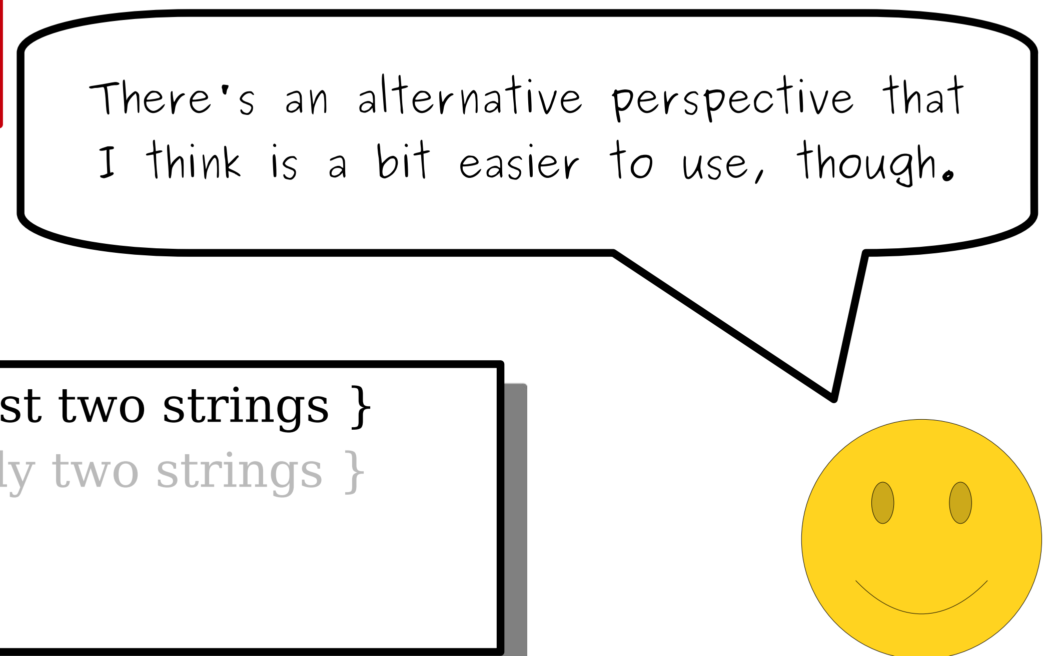
$L_4 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}$ and $n \leq 1000 \}$

RE: Languages with Verifiers

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



R: Languages with Deciders



L_1 = { $\langle M \rangle$ | M is a TM that accepts at least two strings }

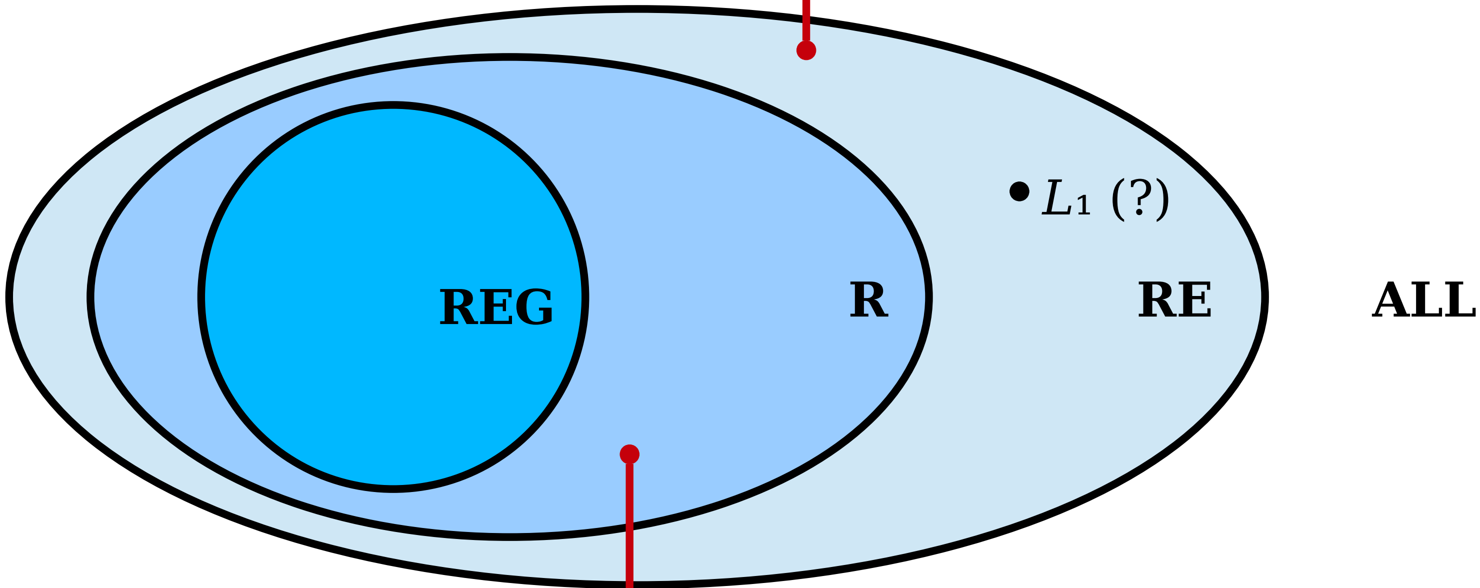
L_2 = { $\langle M \rangle$ | M is a TM that accepts exactly two strings }

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

L_4 = { $\mathtt{a}^n\mathtt{b}^n$ | $n \in \mathbb{N}$ 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

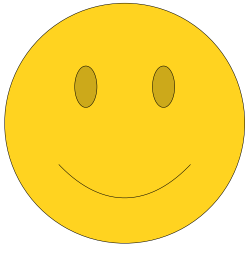
On Problem Set Nine, there's a problem entitled "Double Verification."

L_1 = { ⟨M⟩ | M is a TM that accepts at least two strings }

L_2 = { ⟨M⟩ | M is a TM that accepts exactly two strings }

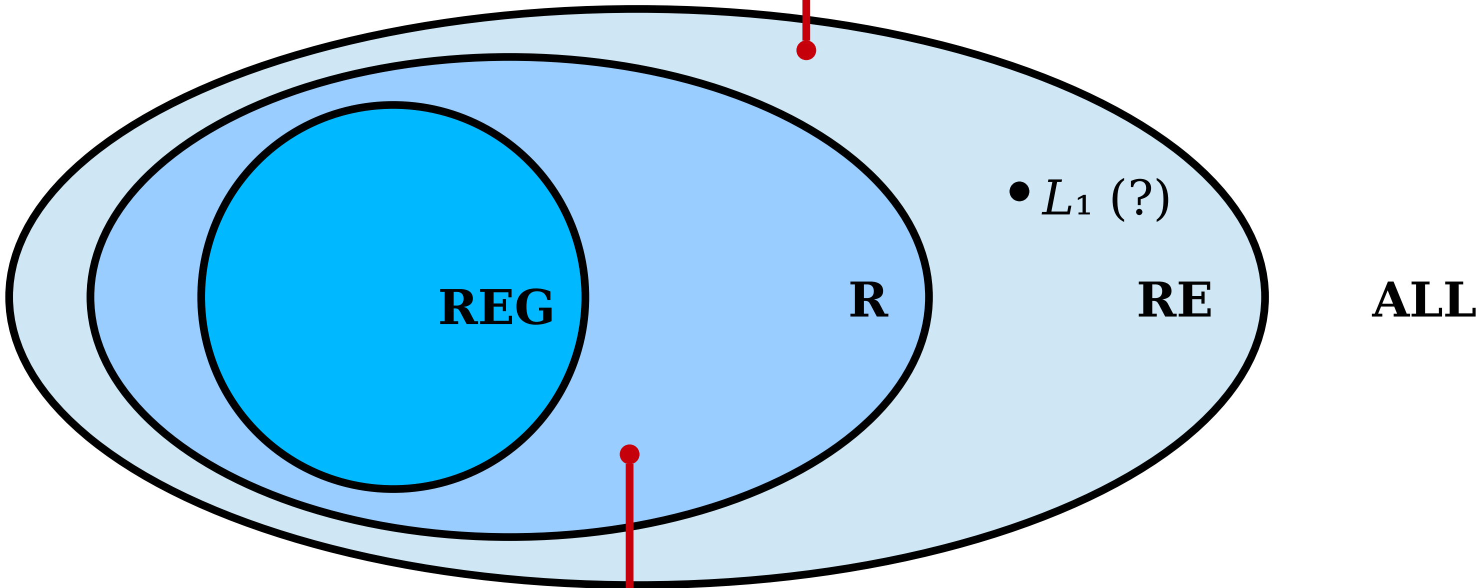
L_3 = { $\mathbf{a}^n\mathbf{b}^n$ | $n \in \mathbb{N}$ and $n > 1000$ }

L_4 = { $\mathbf{a}^n\mathbf{b}^n$ | $n \in \mathbb{N}$ 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

We asked you to prove this statement:

If $L \in$ **RE** and $\overline{L} \in$ **RE**, then $L \in$ **R**.

What exactly does that mean?

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

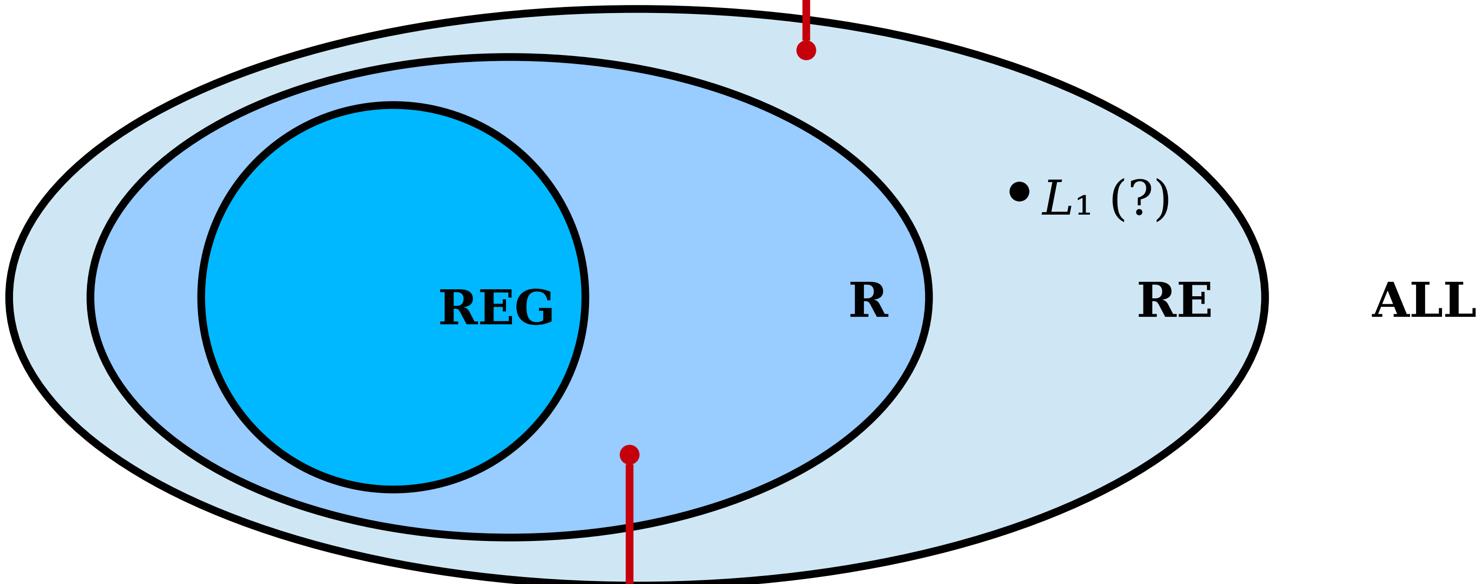
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

$L_4 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}$ 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

If $L \in$ **RE** and $\overline{L} \in$ **RE**, then $L \in$ **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 **RE**.

L_1 = { ⟨M⟩ | M is a TM that accepts at least two strings }

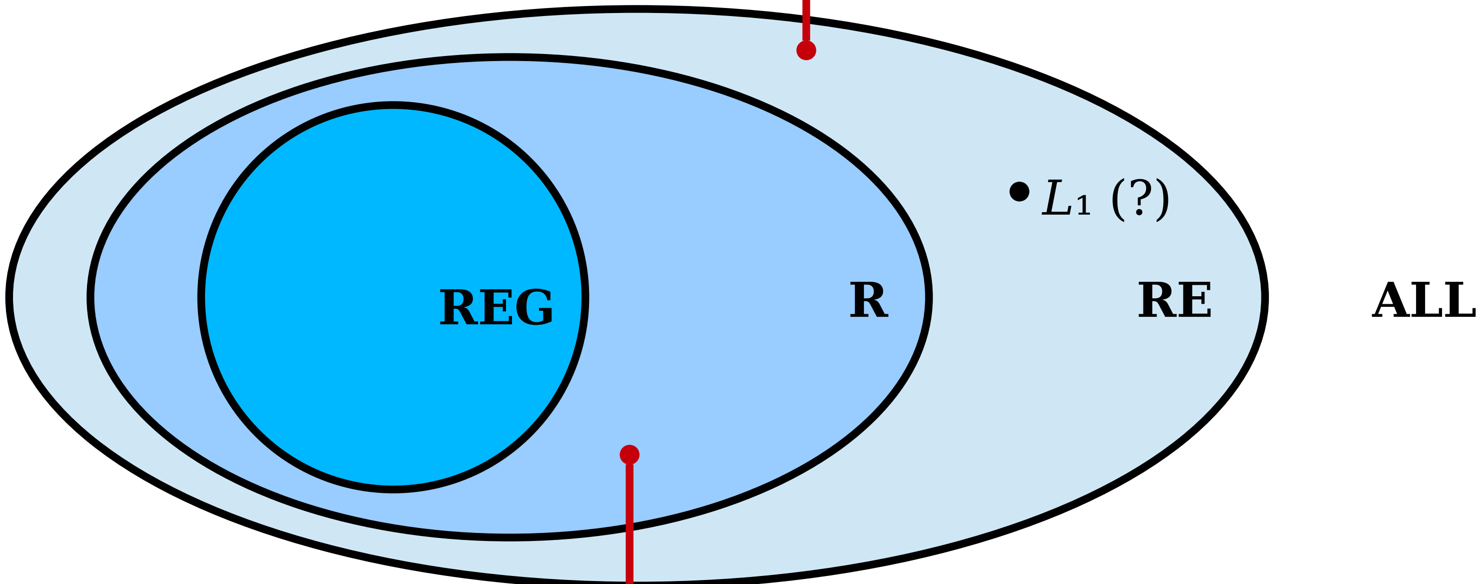
L_2 = { ⟨M⟩ | M is a TM that accepts exactly two strings }

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

L_4 = { $\mathtt{a}^n\mathtt{b}^n$ | $n \in \mathbb{N}$ 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

If $L \in$ **RE** and $\overline{L} \in$ **RE**, then $L \in$ **R**.

But what exactly does it mean for the <u>complement</u> of L to be in **RE**?

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

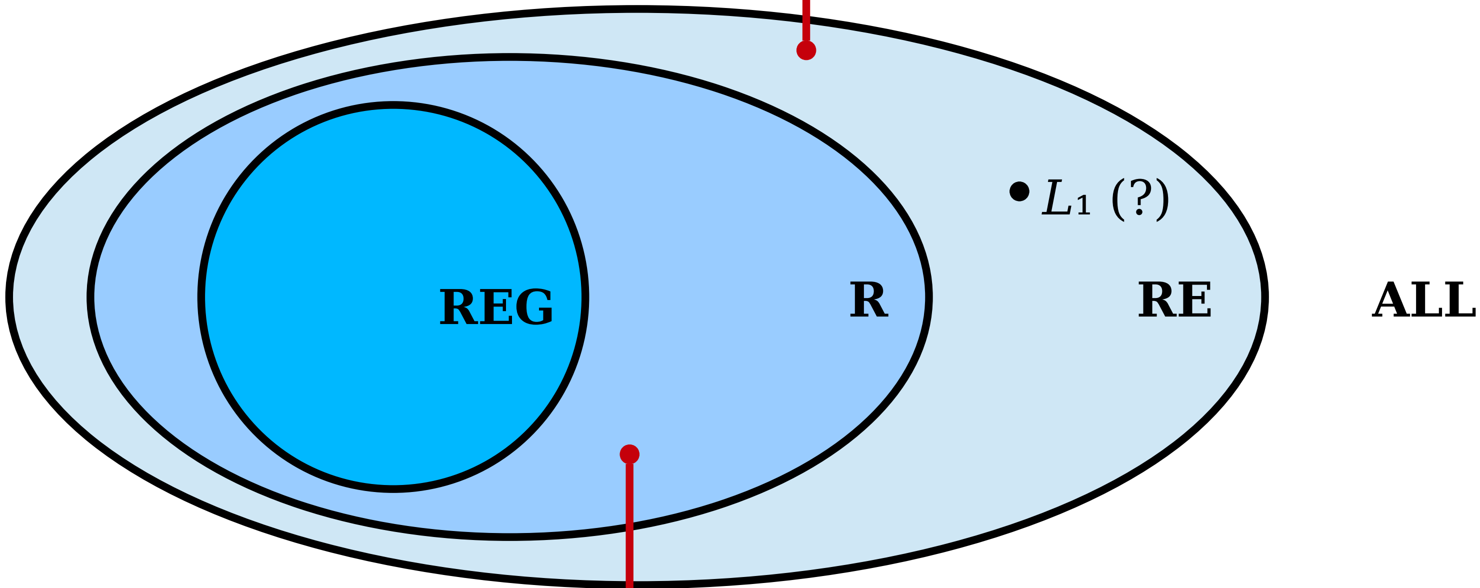
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

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

RE: Languages with Verifiers

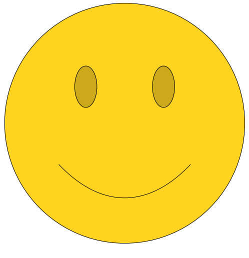
Given any string $w \in L$, could you ***prove*** that $w \in L$?



R: Languages with Deciders

If $L \in$ **RE** and $\overline{L} \in$ **RE**, then $L \in$ **R**.

Going off of our proof-based intuition, if the complement of L is in **RE**, it means that given any string w that is <u>not</u> in L, there's a way to prove it's not in L.



L_1 = { ⟨M⟩ | M is a TM that accepts at least two strings }

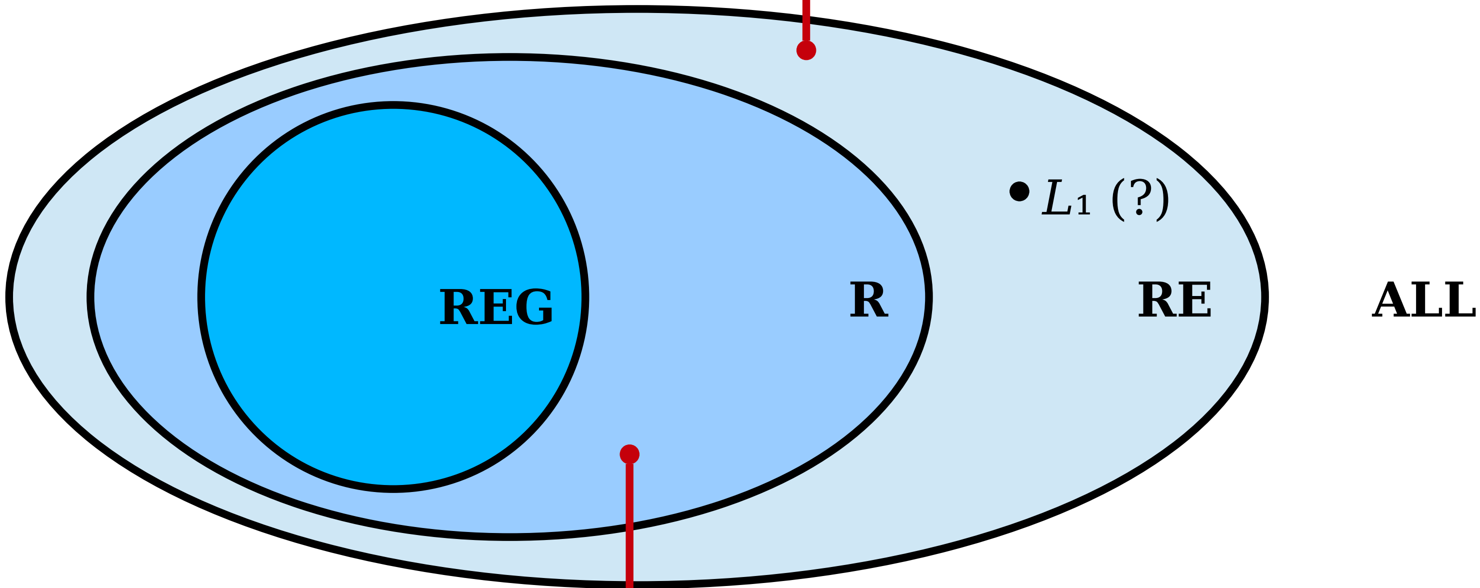
L_2 = { ⟨M⟩ | M is a TM that accepts exactly two strings }

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

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

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 $w \notin L$?

This turns out to be a great way of intuiting the class **R**. A language belongs to **R** if it's in **RE**, and for any string that isn't in the language, there's a way to prove it's not in the language.

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

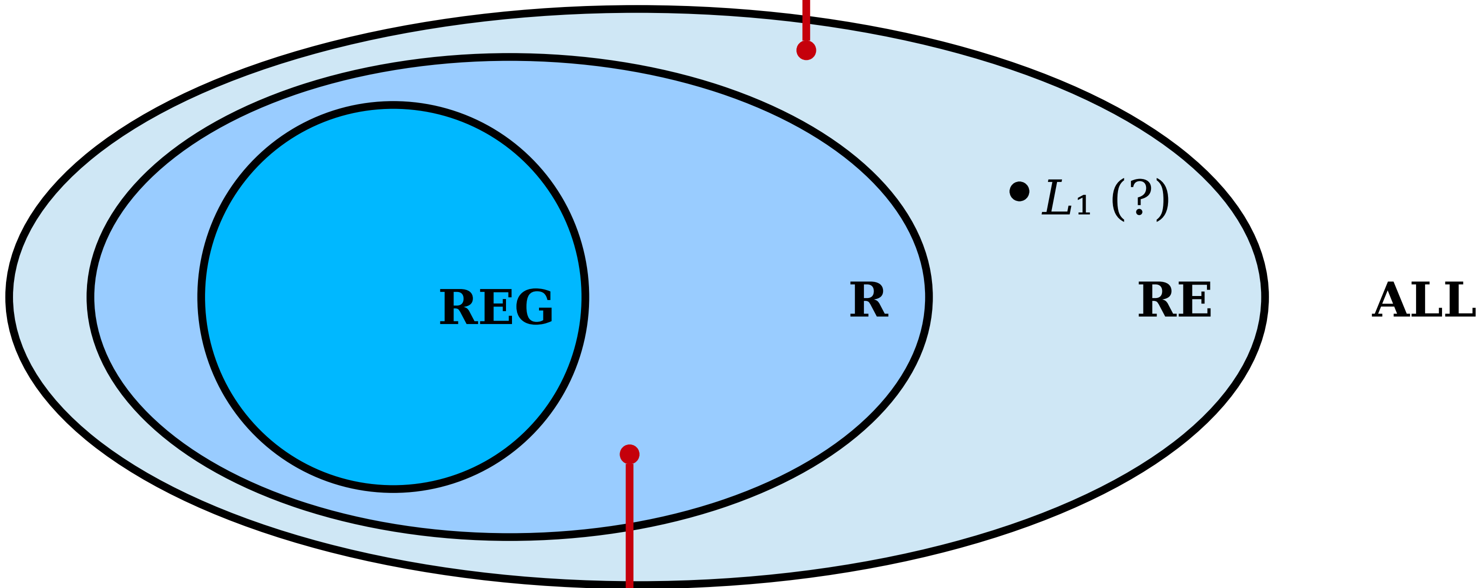
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

$L_4 = \{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N}$ 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 $w \notin L$?

(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 **R**!)

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

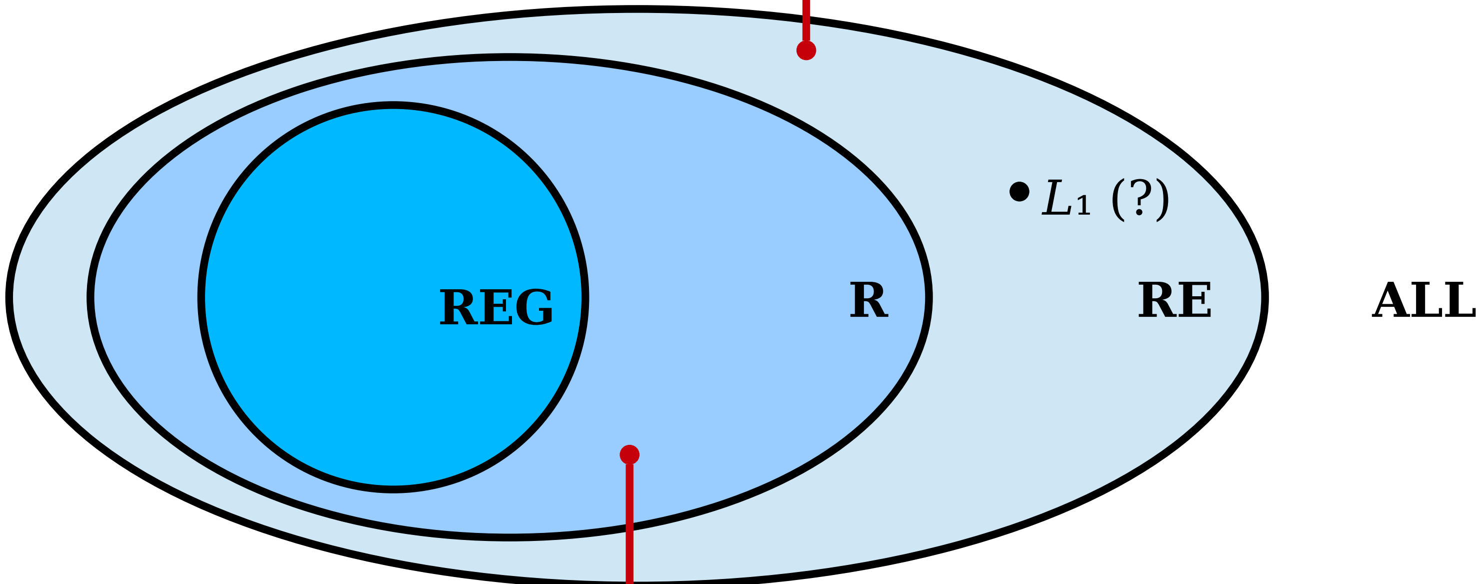
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

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

RE: Languages with Verifiers

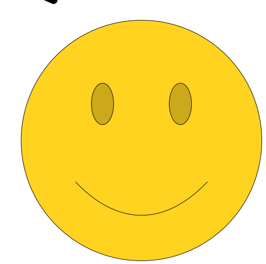
Given any string $w \in L$, could you ***prove*** that $w \in L$?



R: Languages with Deciders

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

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**.



L_1 = { ⟨M⟩ | M is a TM that accepts at least two strings }

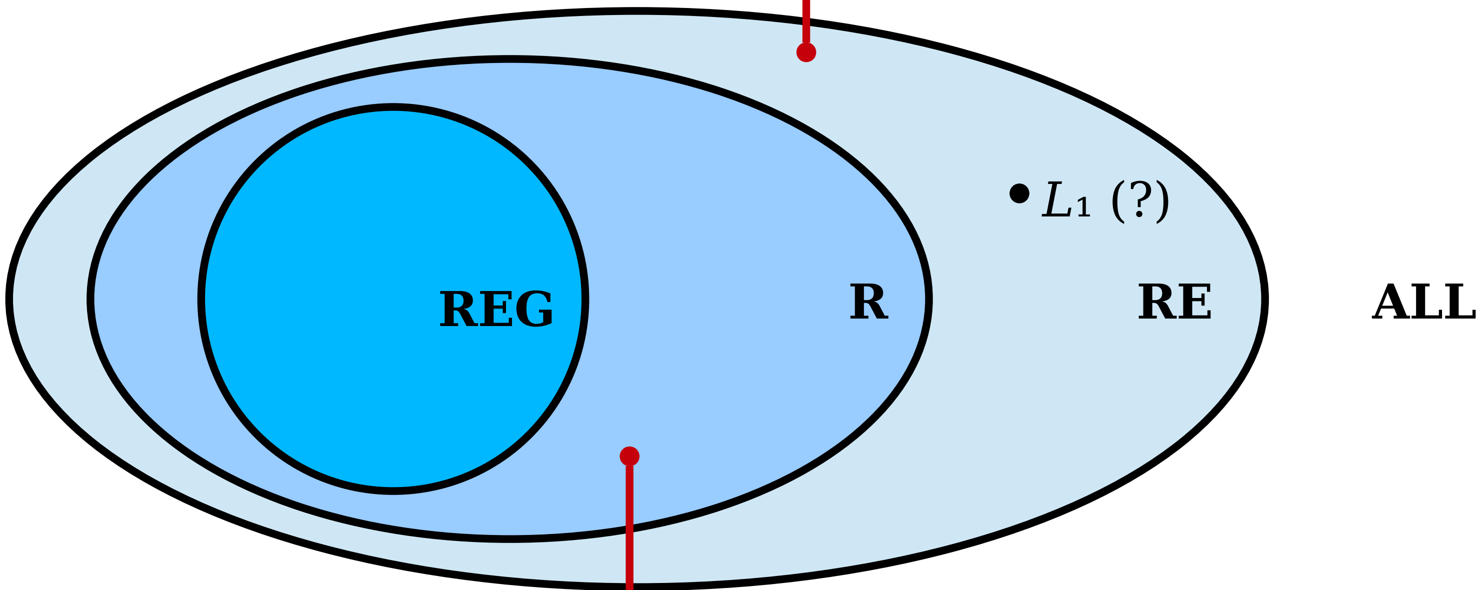
L_2 = { ⟨M⟩ | M is a TM that accepts exactly two strings }

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

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

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 imagine that you have some string that isn't in L_1.

L_1 = { ⟨M⟩ | M is a TM that accepts at least two strings }

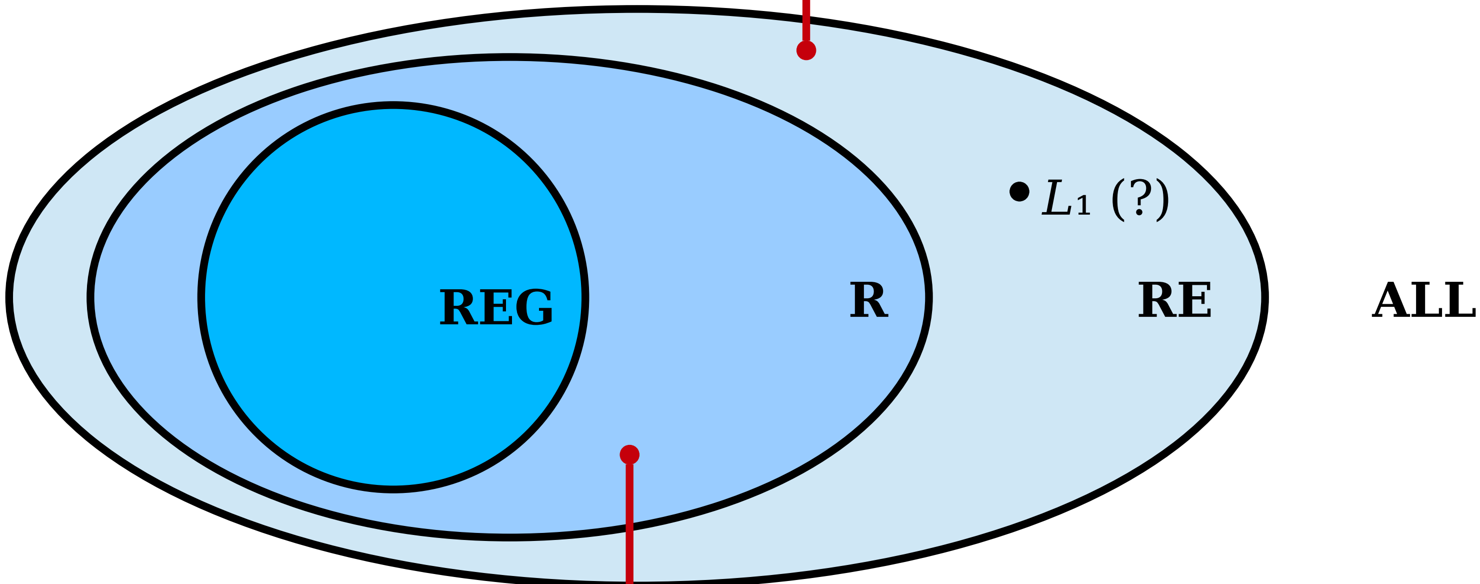
L_2 = { ⟨M⟩ | M is a TM that accepts exactly two strings }

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

L_4 = { $\mathbf{a}^n\mathbf{b}^n$ | $n \in \mathbb{N}$ 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 $w \notin L$?

In other words, imagine you have TM M where it's not the case that M accepts at least two strings.

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

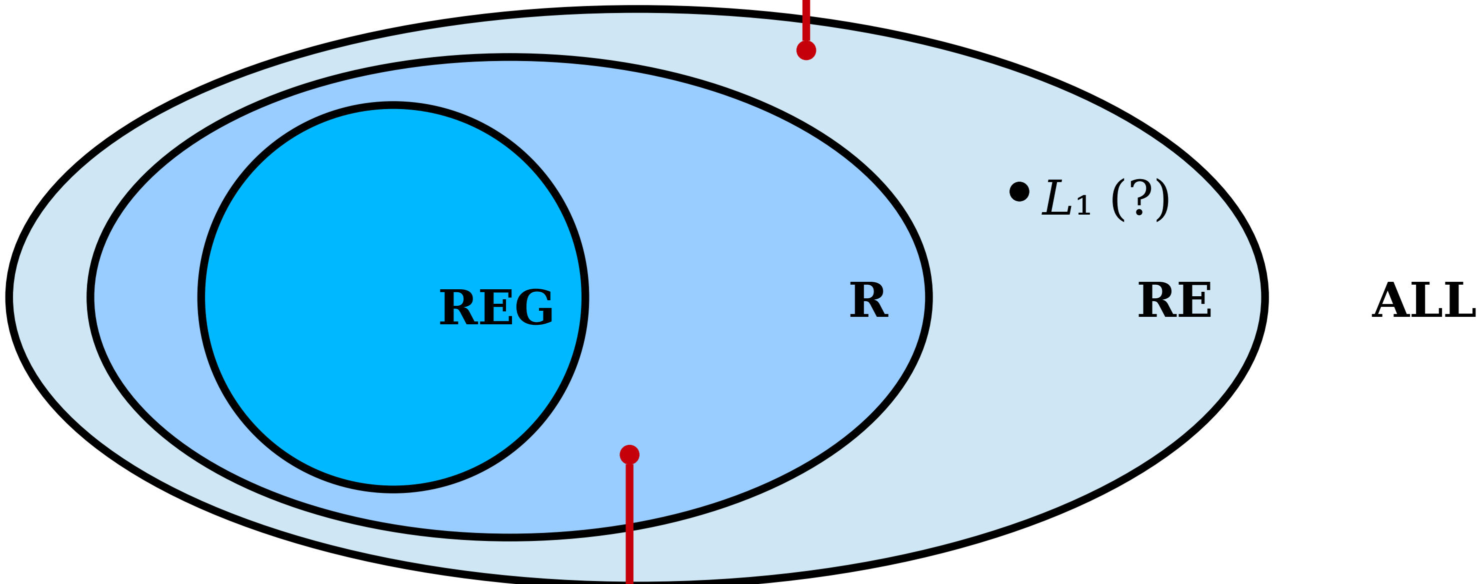
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

$L_4 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}$ 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 $w \notin L$?

That means that M must accept either no strings at all or just one string.

(Do you see why?)

L_1 = { ⟨M⟩ | M is a TM that accepts at least two strings }

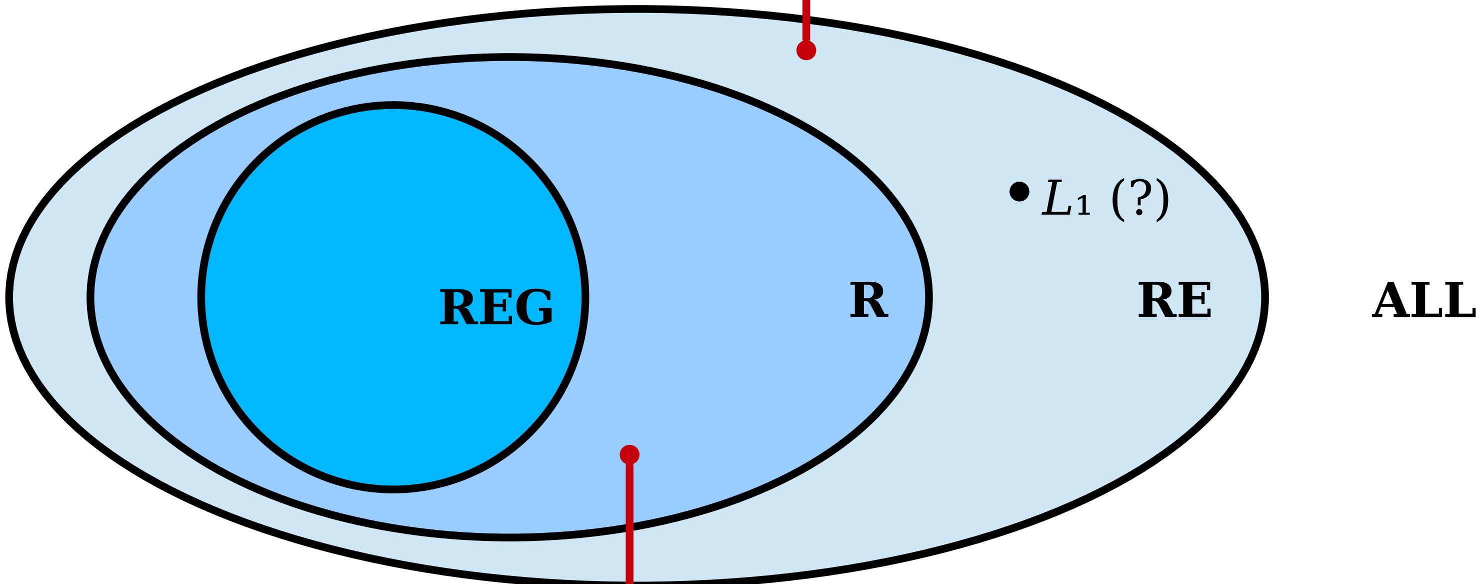
L_2 = { ⟨M⟩ | M is a TM that accepts exactly two strings }

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

L_4 = { $\mathtt{a}^n\mathtt{b}^n$ | $n \in \mathbb{N}$ 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 $w \notin L$?

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?

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

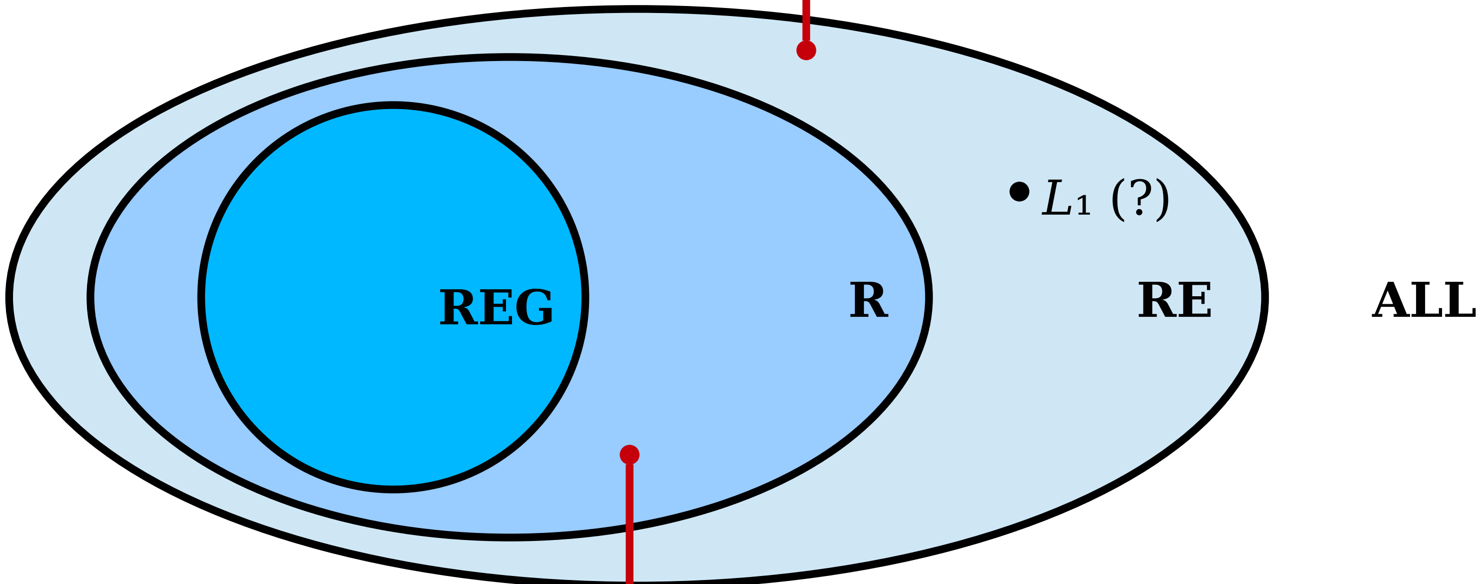
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

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

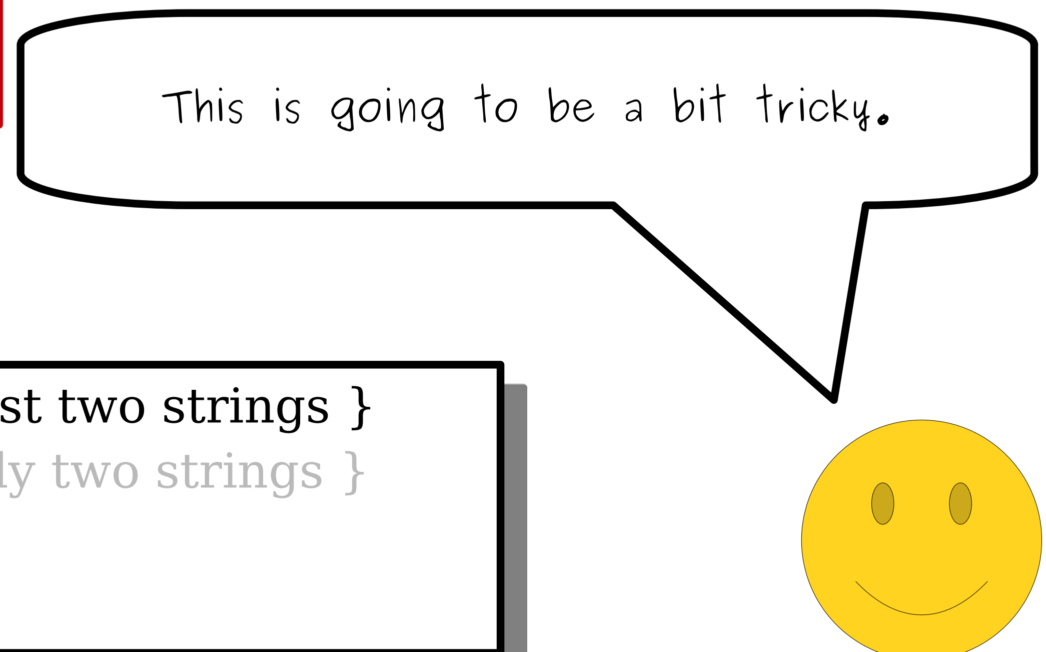
RE: Languages with Verifiers

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 $w \notin L$?



$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

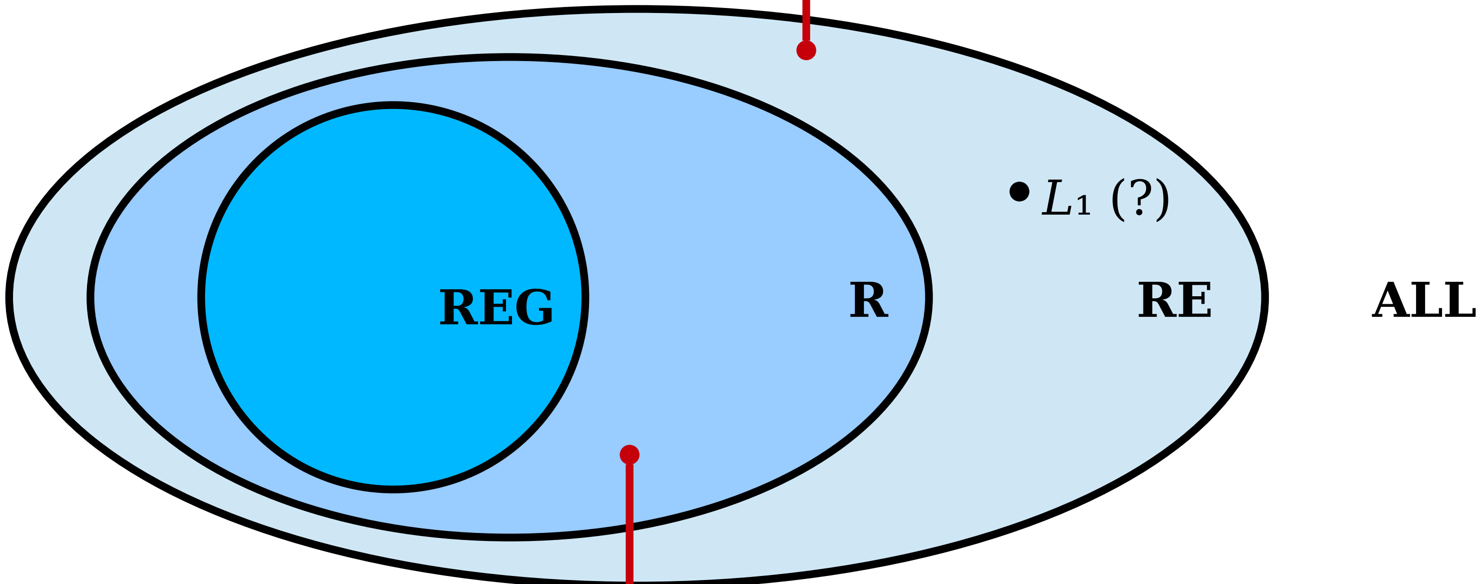
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

$L_4 = \{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N}$ 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 $w \notin L$?

If you want to convince someone that M only accepts at most one string, you need to convince them that out of the infinitely many strings that are out there, the TM accepts at most one.

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

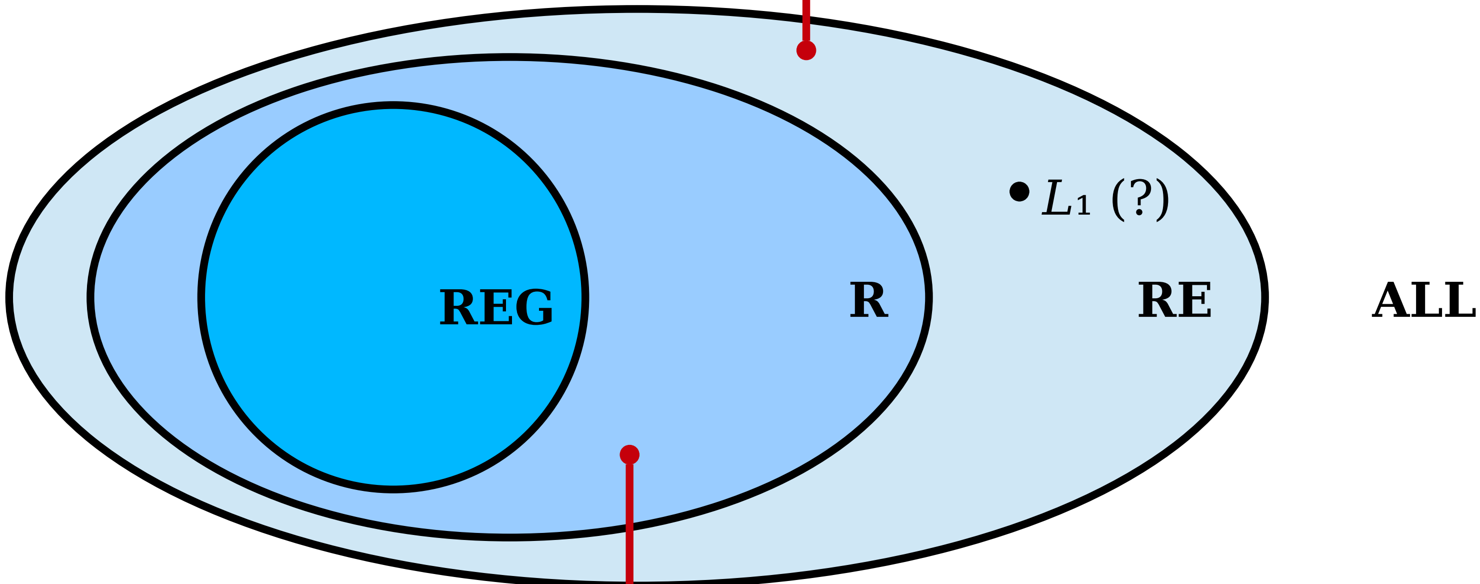
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

$L_4 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}$ 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 $w \notin L$?

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.

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

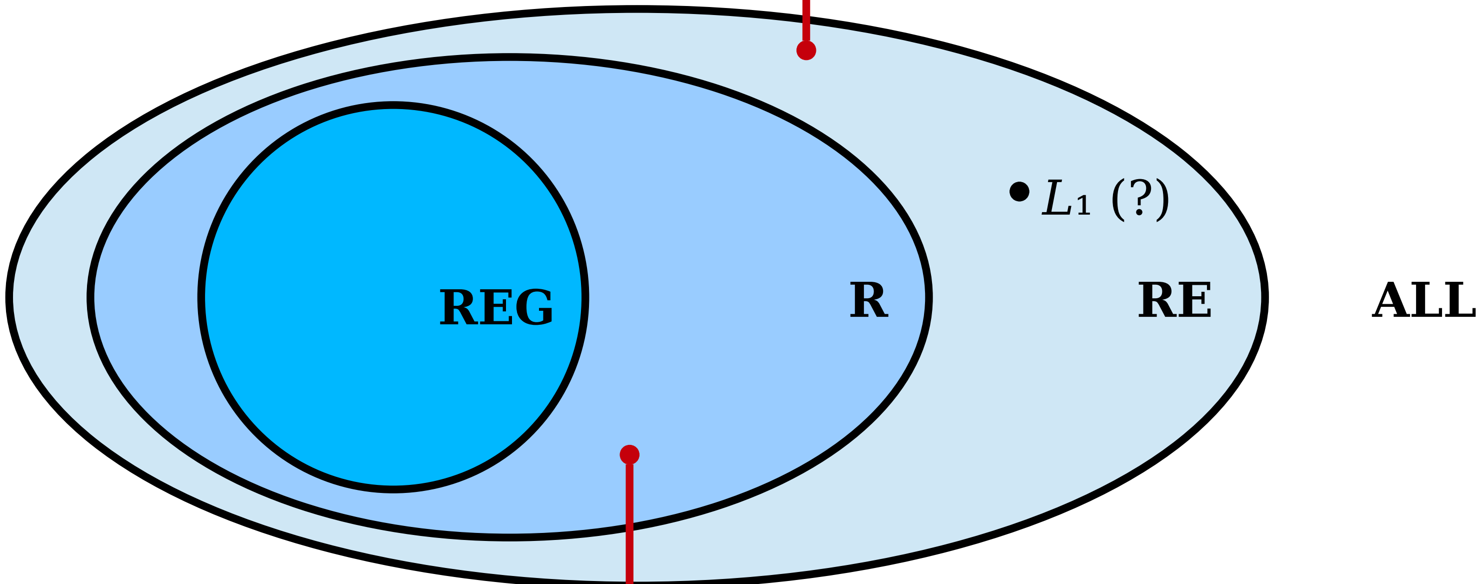
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

$L_4 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}$ 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 $w \notin L$?

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…

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

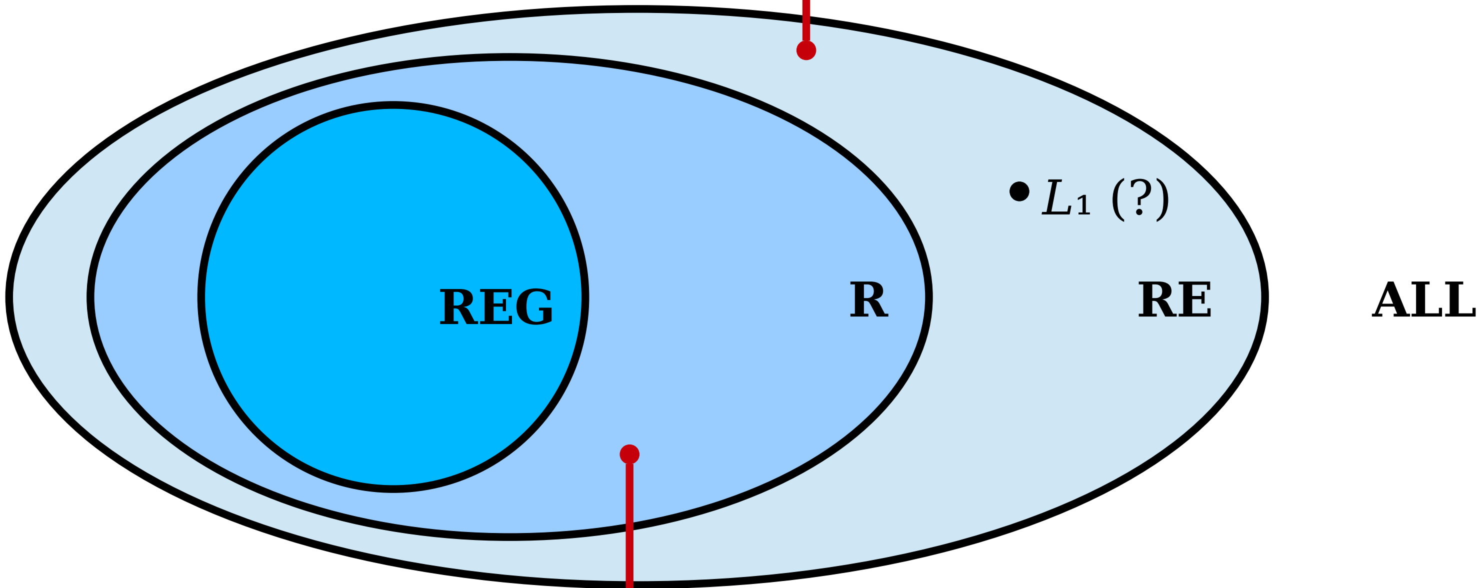
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

$L_4 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}$ 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 = { ⟨M⟩ | M is a TM that accepts at least two strings }

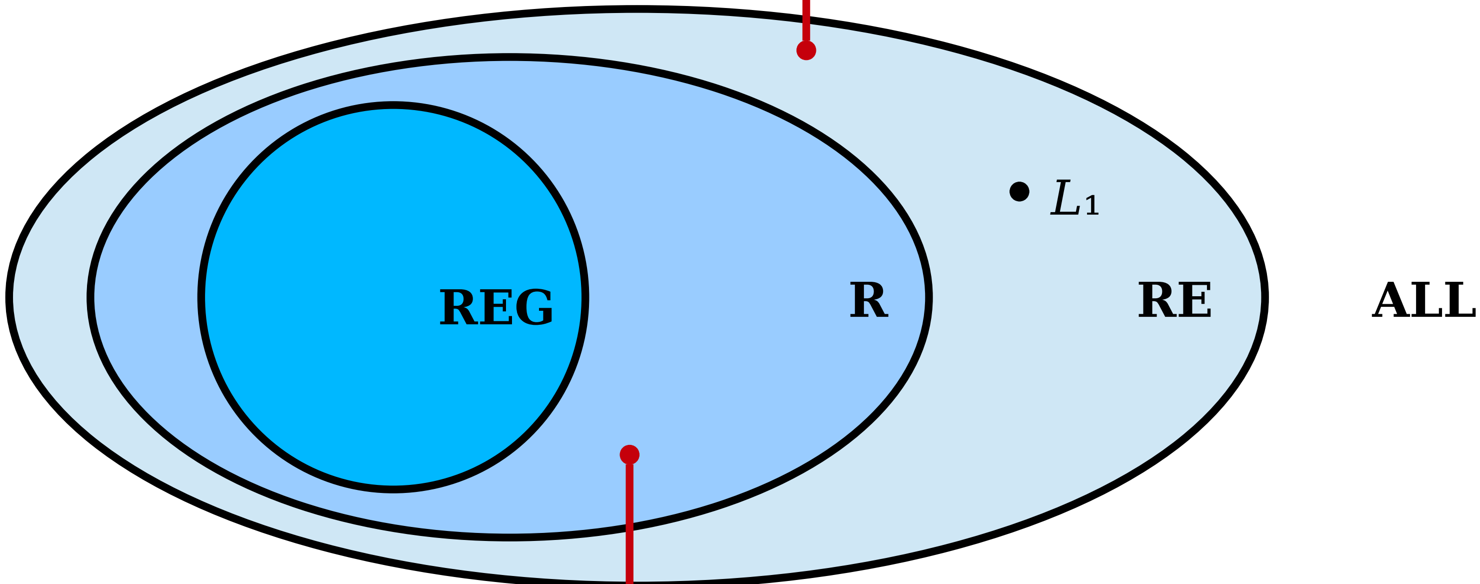
L_2 = { ⟨M⟩ | M is a TM that accepts exactly two strings }

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

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

RE: Languages with Verifiers

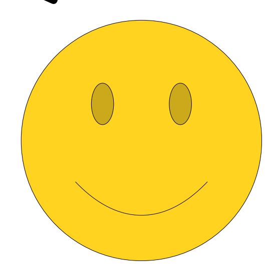
Given any string $w \in L$, could you ***prove*** that $w \in L$?



R: Languages with Deciders

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

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



L_1 = { ⟨M⟩ | M is a TM that accepts at least two strings }

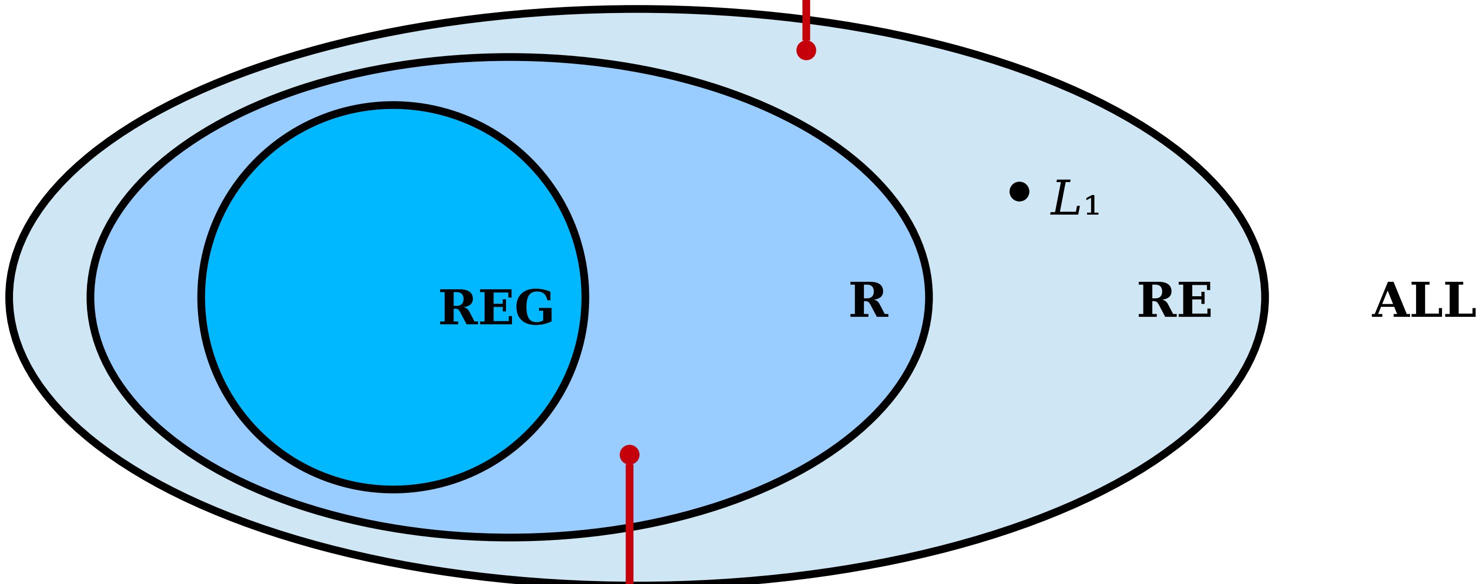
L_2 = { ⟨M⟩ | M is a TM that accepts exactly two strings }

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

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

RE: Languages with Verifiers

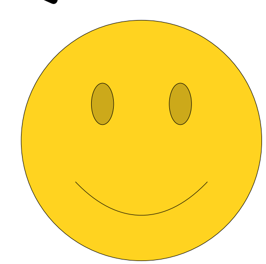
Given any string $w \in L$, could you ***prove*** that $w \in L$?



R: Languages with Deciders

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

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 = { ⟨M⟩ | M is a TM that accepts at least two strings }

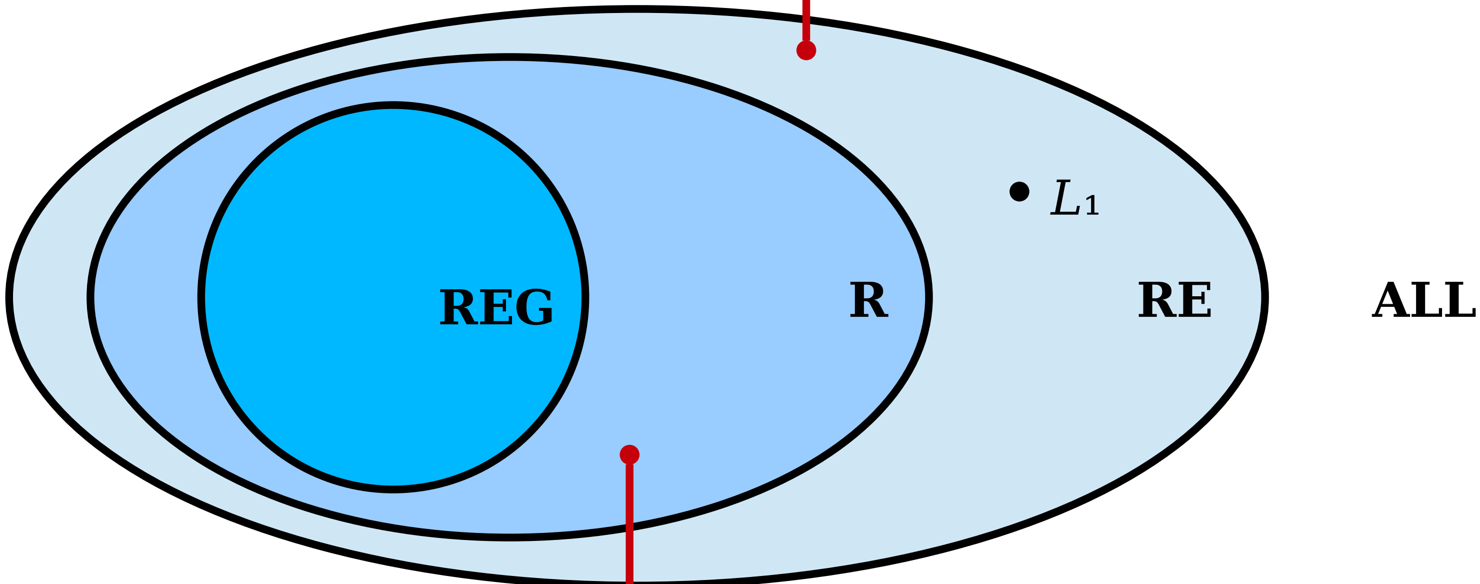
L_2 = { ⟨M⟩ | M is a TM that accepts exactly two strings }

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

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

RE: Languages with Verifiers

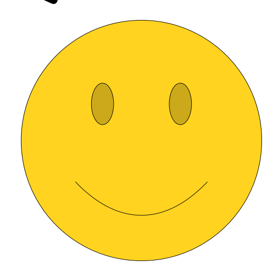
Given any string $w \in L$, could you ***prove*** that $w \in L$?



R: Languages with Deciders

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

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 = { ⟨M⟩ | M is a TM that accepts at least two strings }

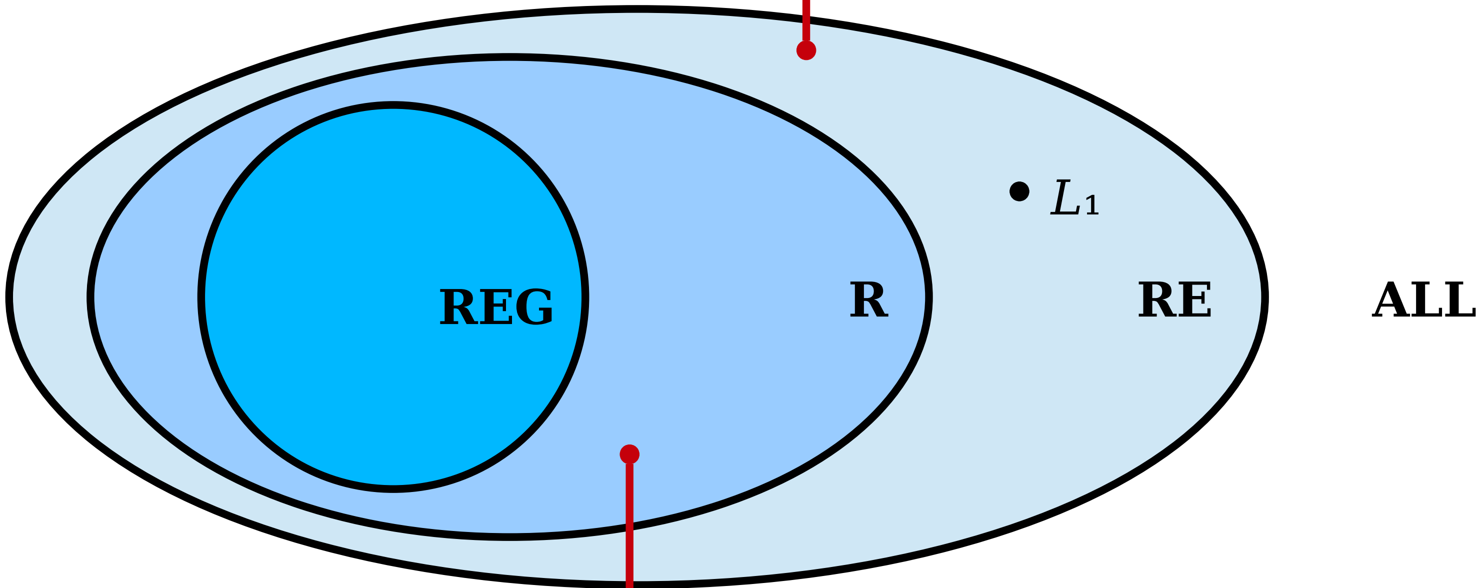
L_2 = { ⟨M⟩ | M is a TM that accepts exactly two strings }

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

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

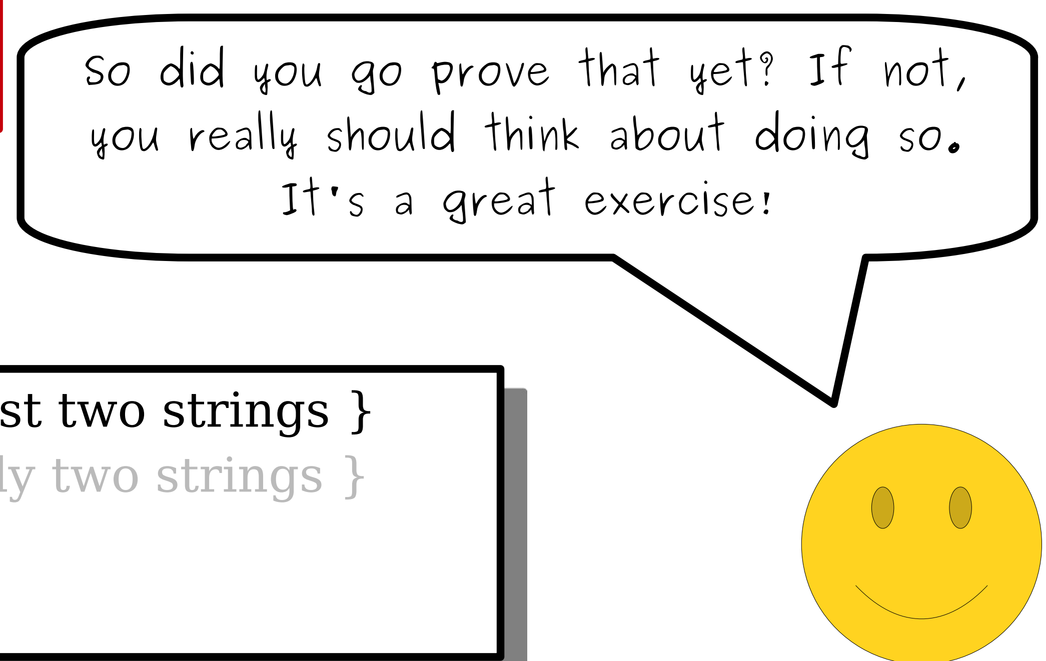
RE: Languages with Verifiers

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 $w \notin L$?



$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

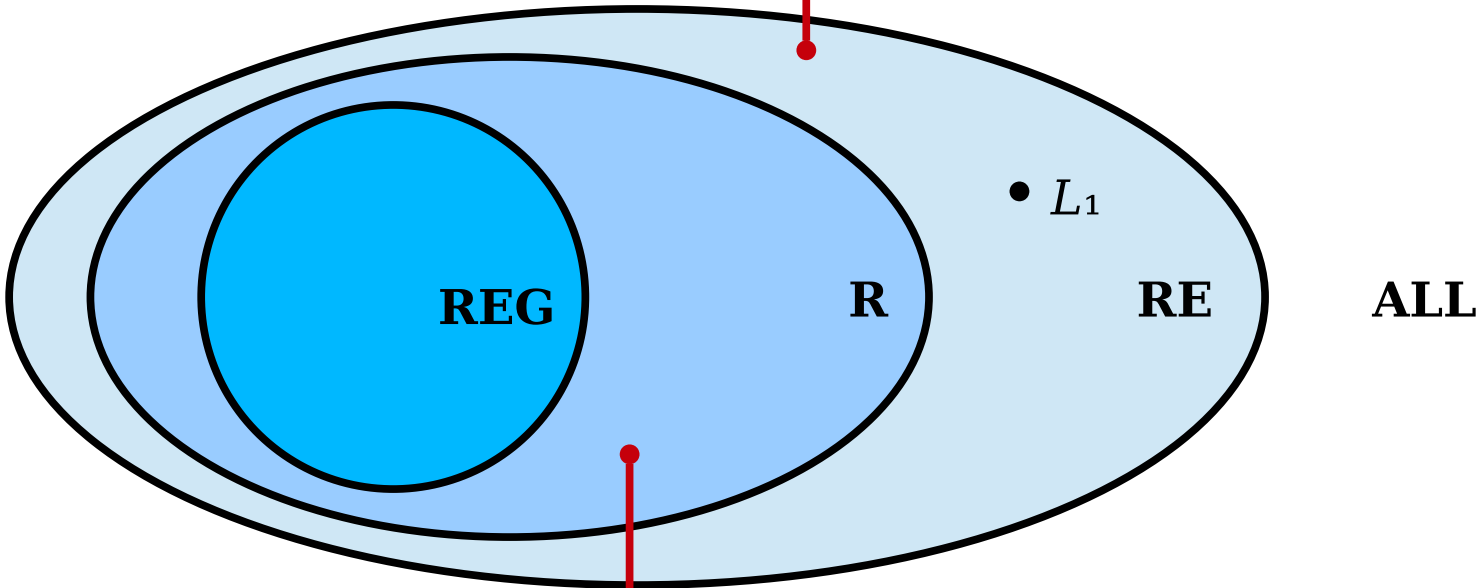
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

$L_4 = \{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N}$ 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 $w \notin L$?

So at this point we've got this language settled in the right place. It's in **RE**, but it's not in **R**.

L_1 = { ⟨M⟩ | M is a TM that accepts at least two strings }

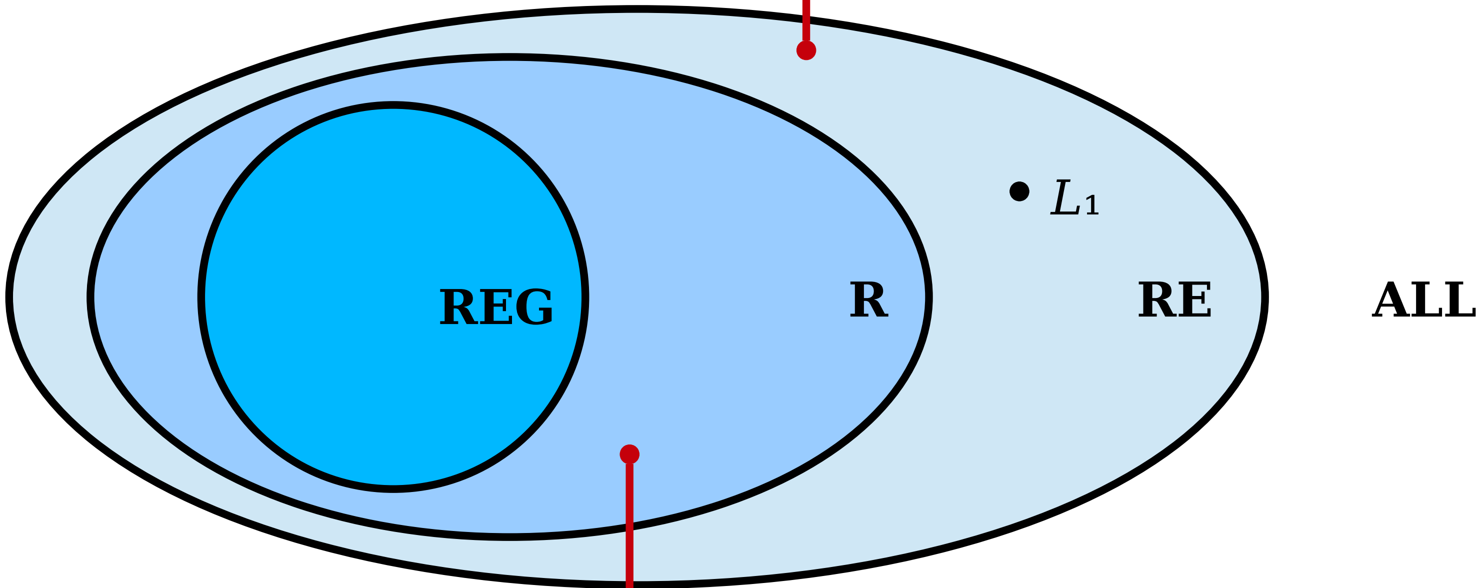
L_2 = { ⟨M⟩ | M is a TM that accepts exactly two strings }

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

L_4 = { $\mathtt{a}^n\mathtt{b}^n$ | $n \in \mathbb{N}$ 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 $w \notin L$?

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.

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

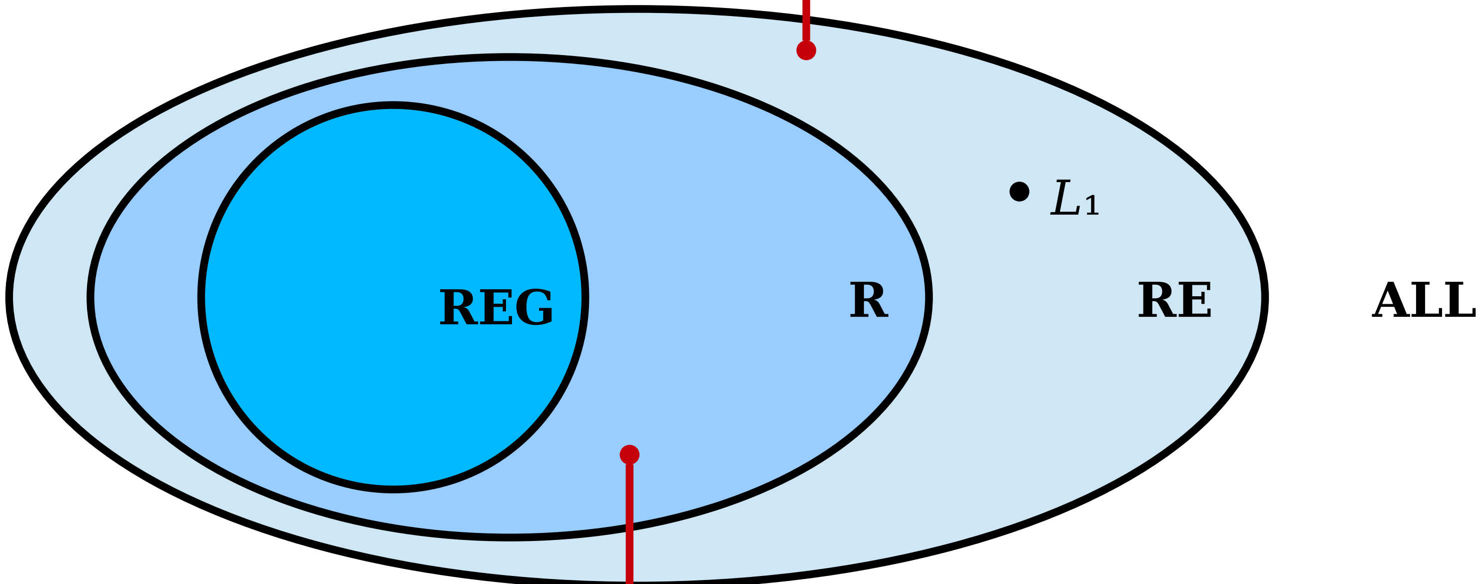
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

$L_4 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}$ 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 $w \notin L$?

If you look at the description of the language, you can see that it says something about TMs that accept at least two strings.

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

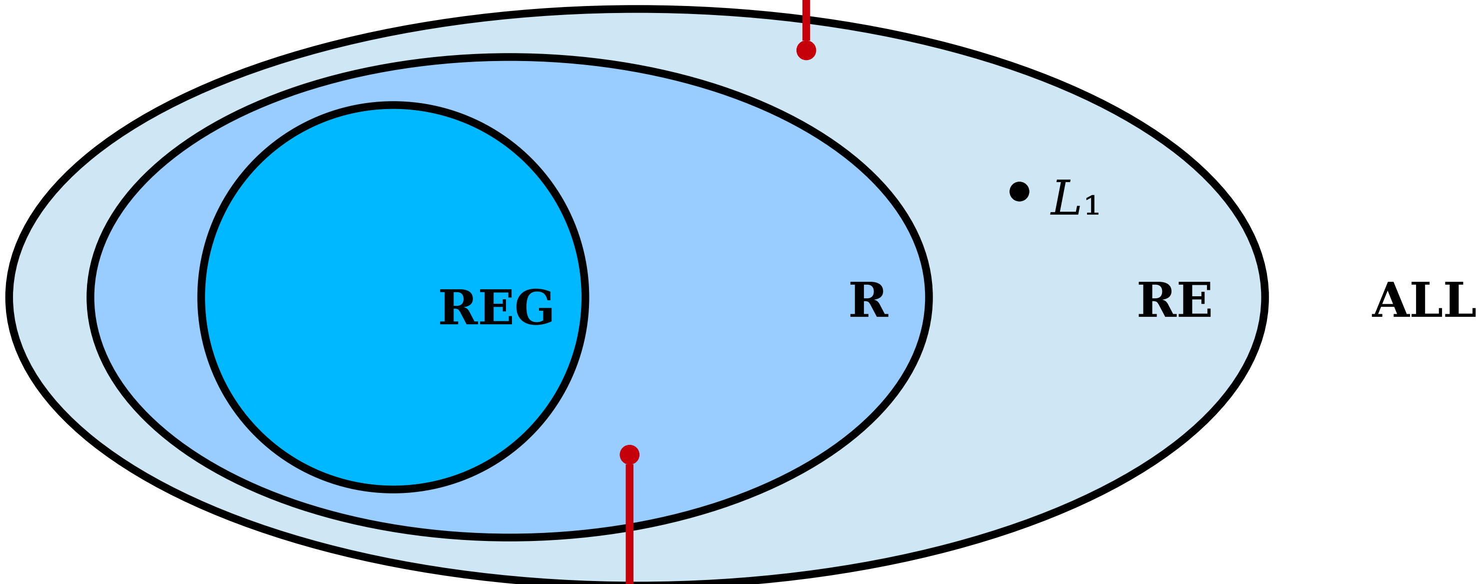
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

$L_4 = \{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N}$ 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 $w \notin L$?

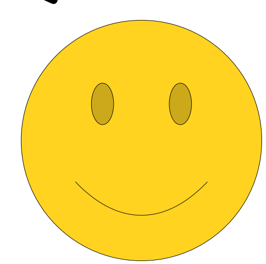
A lot of people ask – "Isn't it really easy to build a TM that accepts at least two strings? So shouldn't this be decidable? Or even regular?"

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

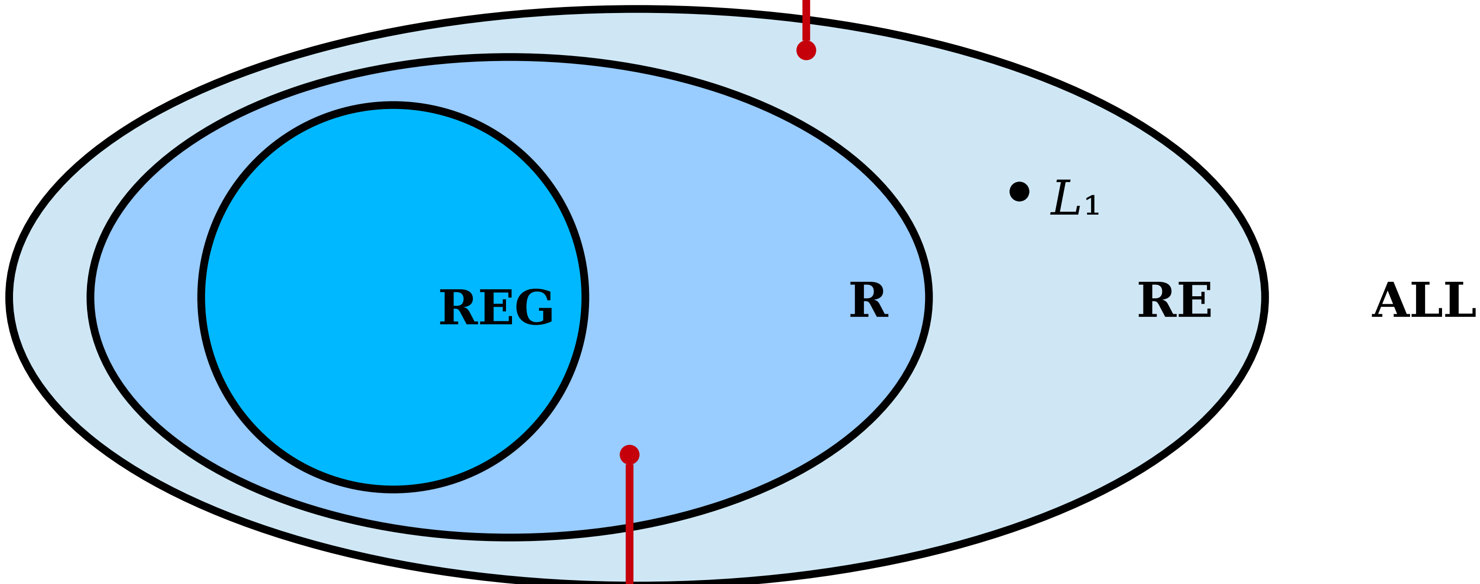
$L_3 = \{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N}$ and $n > 1000 \}$

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



RE: Languages with Verifiers

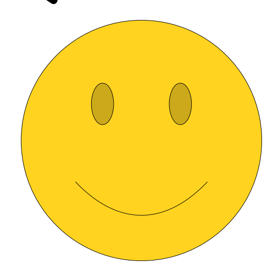
Given any string $w \in L$, could you ***prove*** that $w \in L$?



R: Languages with Deciders

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

The answer to that question is "yes, and no."



L_1 = { ⟨M⟩ | M is a TM that accepts at least two strings }

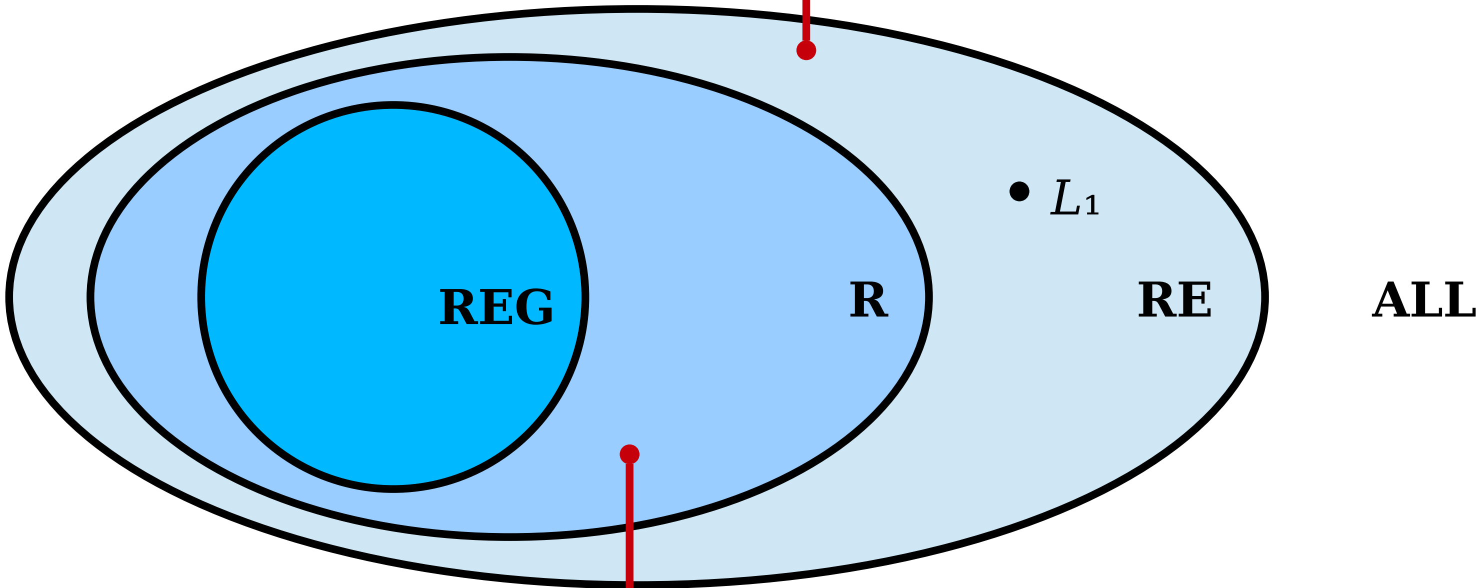
L_2 = { ⟨M⟩ | M is a TM that accepts exactly two strings }

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

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

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 = { ⟨M⟩ | M is a TM that accepts at least two strings }

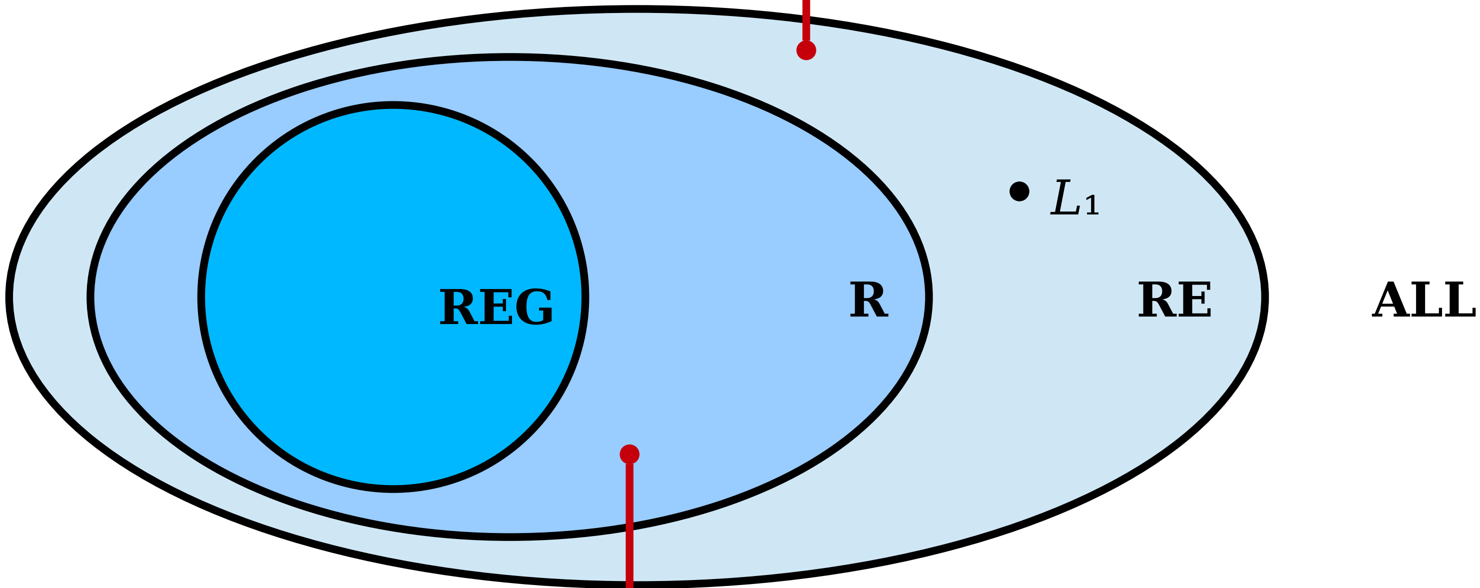
L_2 = { ⟨M⟩ | M is a TM that accepts exactly two strings }

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

L_4 = { $\mathtt{a}^n\mathtt{b}^n$ | $n \in \mathbb{N}$ 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 $w \notin L$?

We can do that by just building a TM that accepts everything, for example.

L_1 = { ⟨M⟩ | M is a TM that accepts at least two strings }

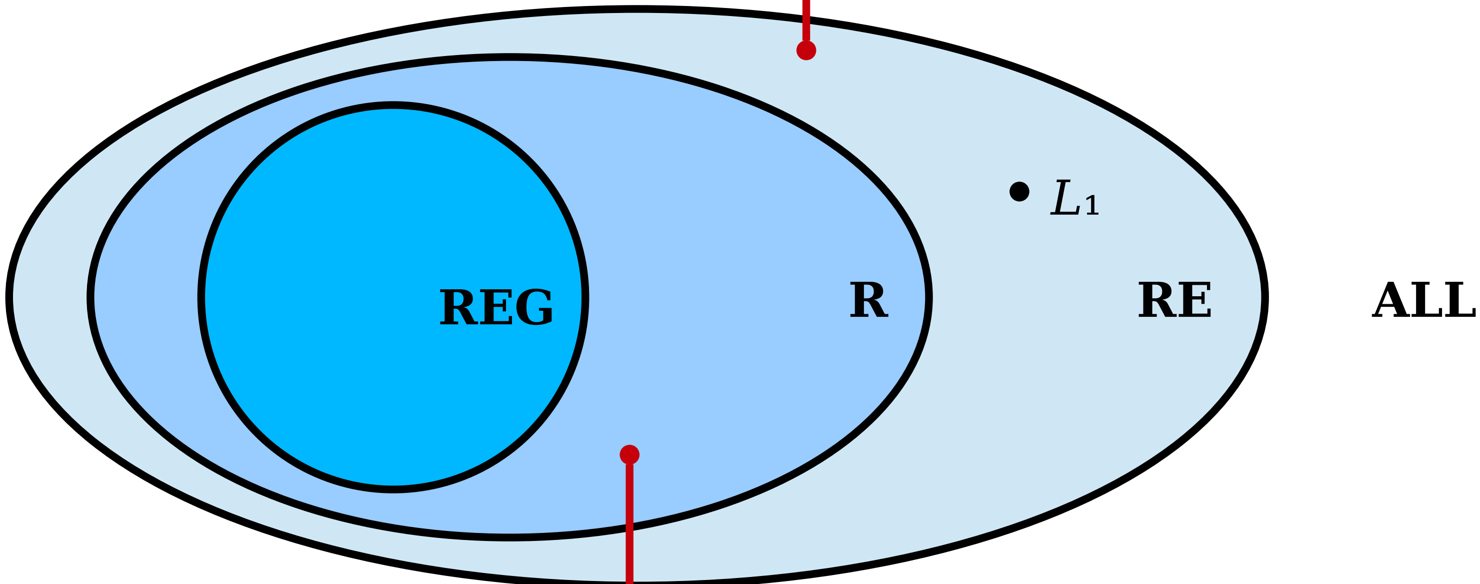
L_2 = { ⟨M⟩ | M is a TM that accepts exactly two strings }

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

L_4 = { $\mathtt{a}^n\mathtt{b}^n$ | $n \in \mathbb{N}$ 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 $w \notin L$?

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.

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

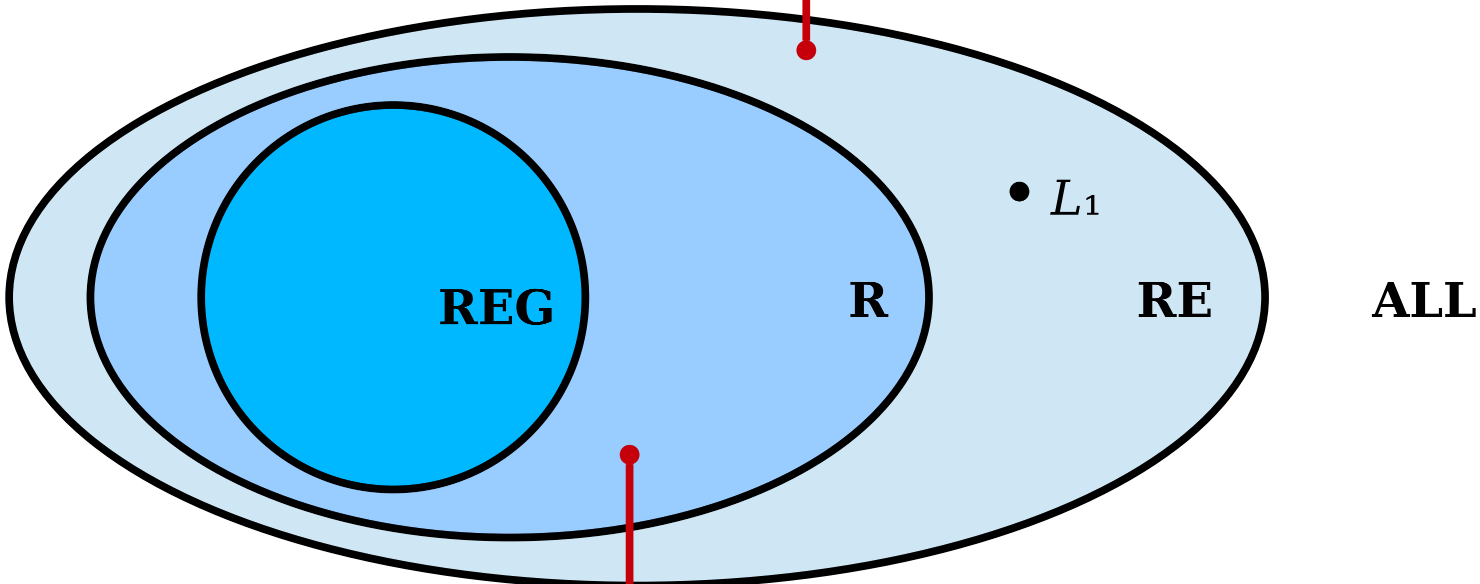
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

$L_4 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}$ 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 $w \notin L$?

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_1 = { ⟨M⟩ | M is a TM that accepts at least two strings }

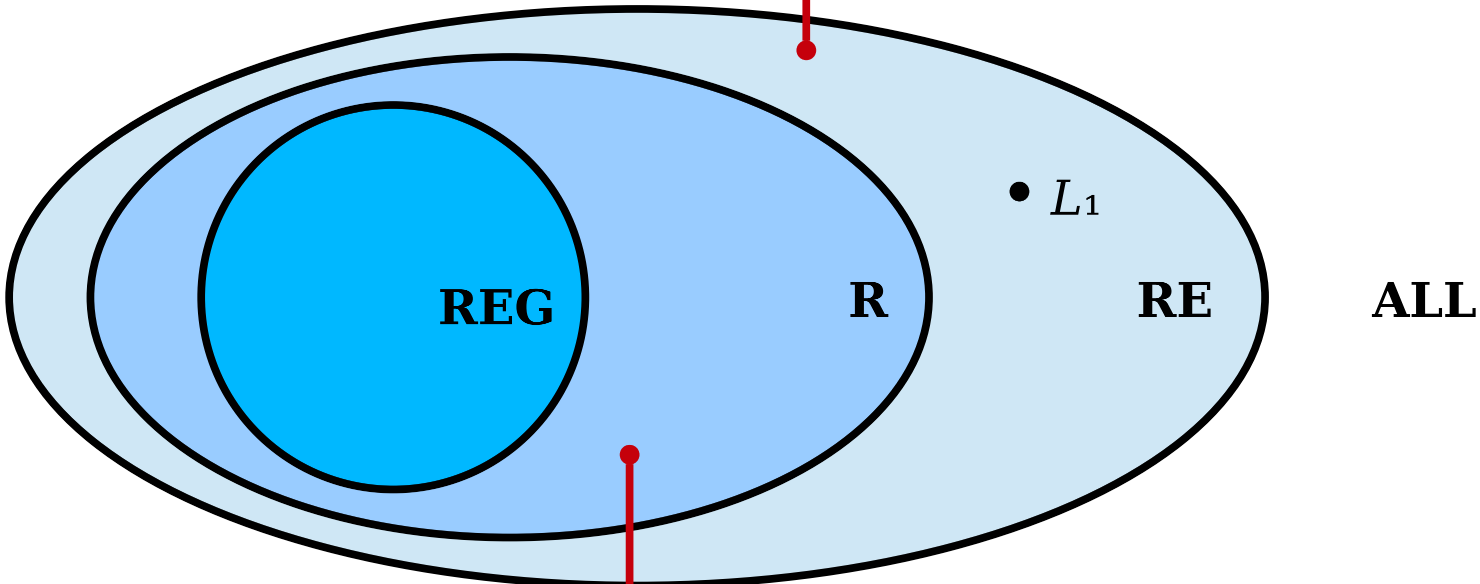
L_2 = { ⟨M⟩ | M is a TM that accepts exactly two strings }

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

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

RE: Languages with Verifiers

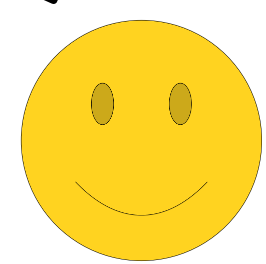
Given any string $w \in L$, could you ***prove*** that $w \in L$?



R: Languages with Deciders

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

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.



$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

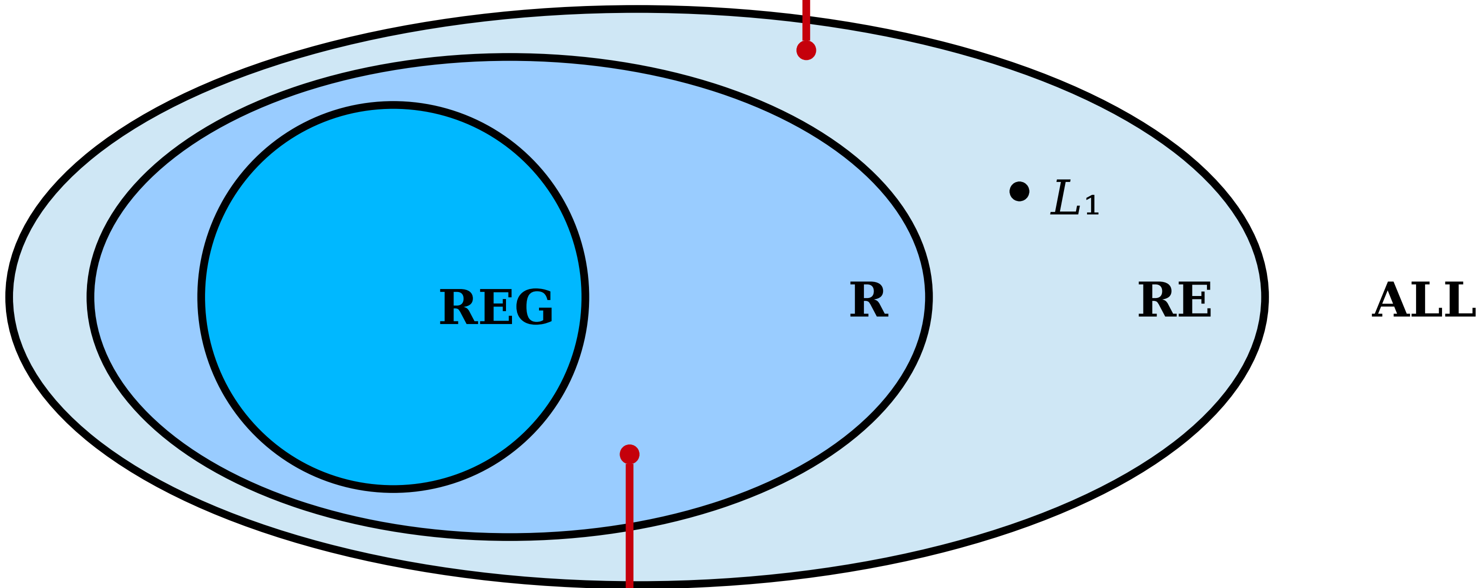
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

$L_4 = \{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N}$ 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 $w \notin L$?

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.

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

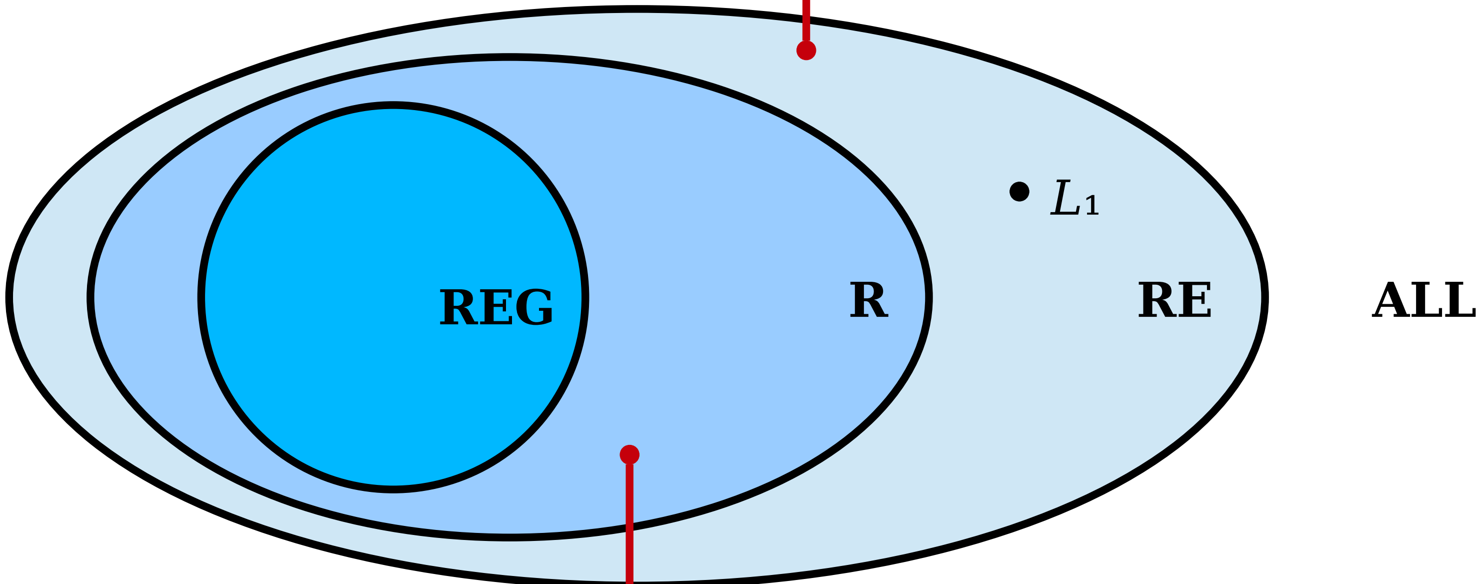
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

$L_4 = \{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N}$ 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 with that said, let's move on to the second language!

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

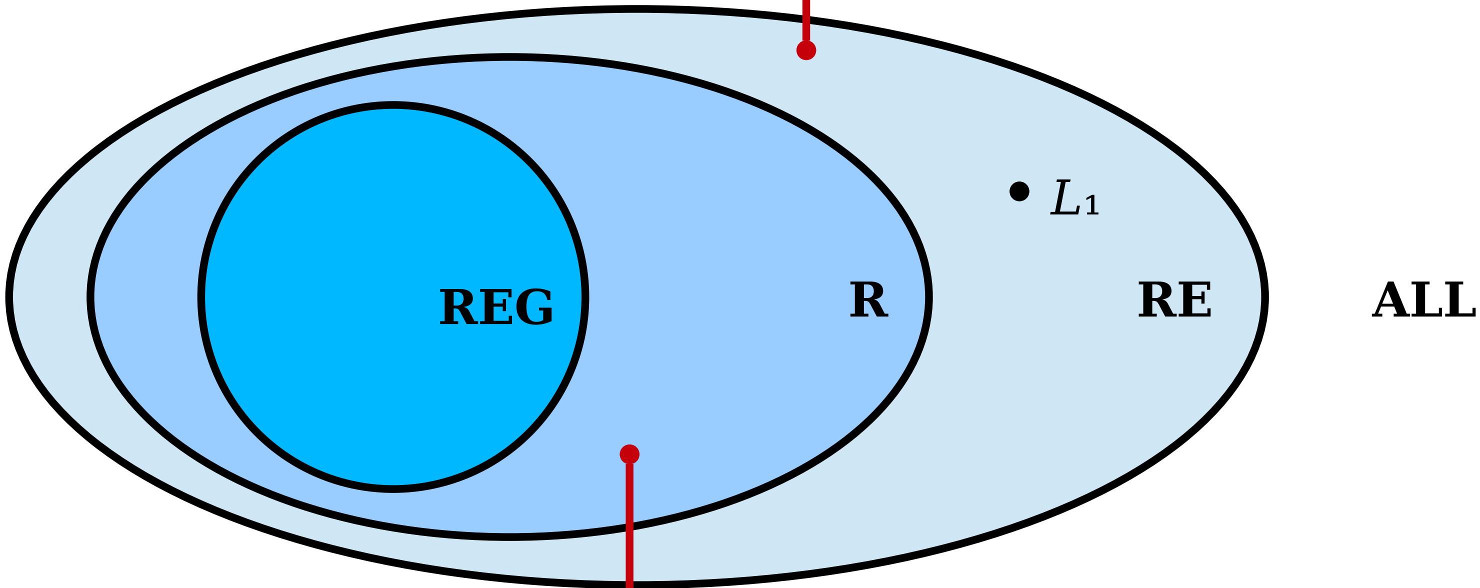
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

$L_4 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}$ 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 $w \notin L$?

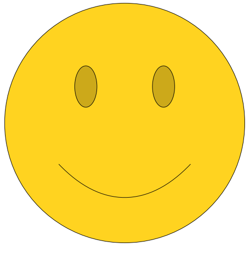
Before I talk about this particular problem, take a few minutes to think about where you believe this should go in the Lava Diagram. Once you've done that, let's rejoin and keep talking.

L_1 = { ⟨M⟩ | M is a TM that accepts at least two strings }

L_2 = { ⟨M⟩ | M is a TM that accepts exactly two strings }

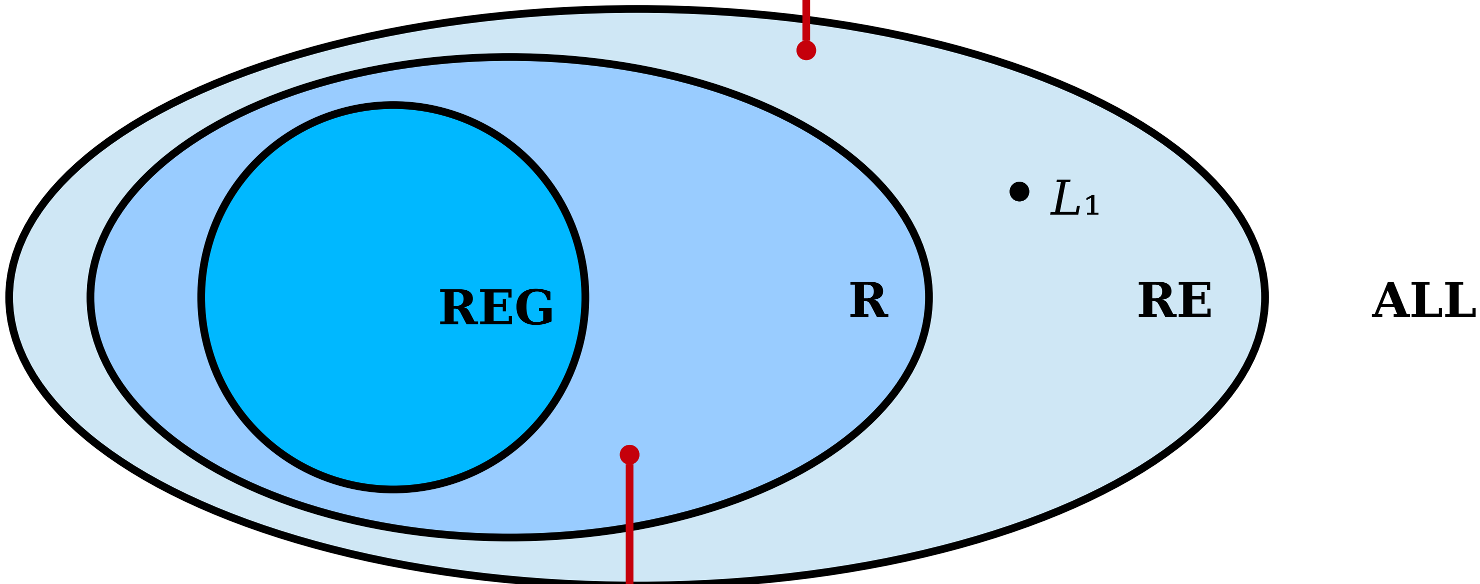
L_3 = { $\mathtt{a}^n\mathtt{b}^n$ | $n \in \mathbb{N}$ and $n > 1000$ }

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



RE: Languages with Verifiers

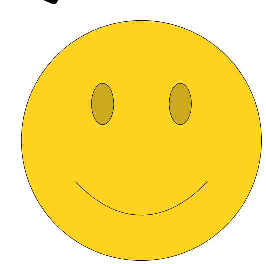
Given any string $w \in L$, could you ***prove*** that $w \in L$?



R: Languages with Deciders

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

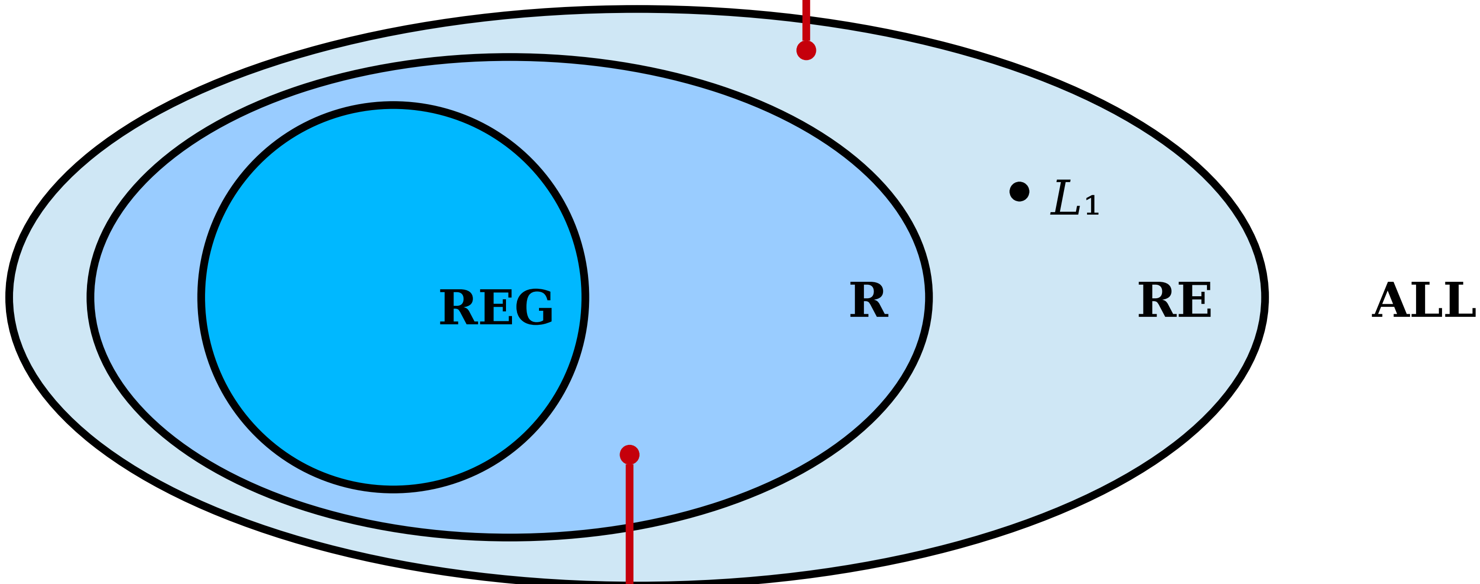
Did you actually go and think about it? If not, you should. Like, seriously. It's good practice.



L_1 = { ⟨M⟩ | M is a TM that accepts at least two strings }
L_2 = { ⟨M⟩ | M is a TM that accepts exactly two strings }
L_3 = { $\mathbf{a}^n\mathbf{b}^n$ | $n \in \mathbb{N}$ and $n > 1000$ }
L_4 = { $\mathbf{a}^n\mathbf{b}^n$ | $n \in \mathbb{N}$ and $n \leq 1000$ }

RE: Languages with Verifiers

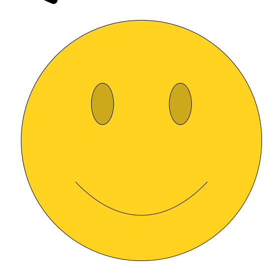
Given any string $w \in L$, could you ***prove*** that $w \in L$?



R: Languages with Deciders

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

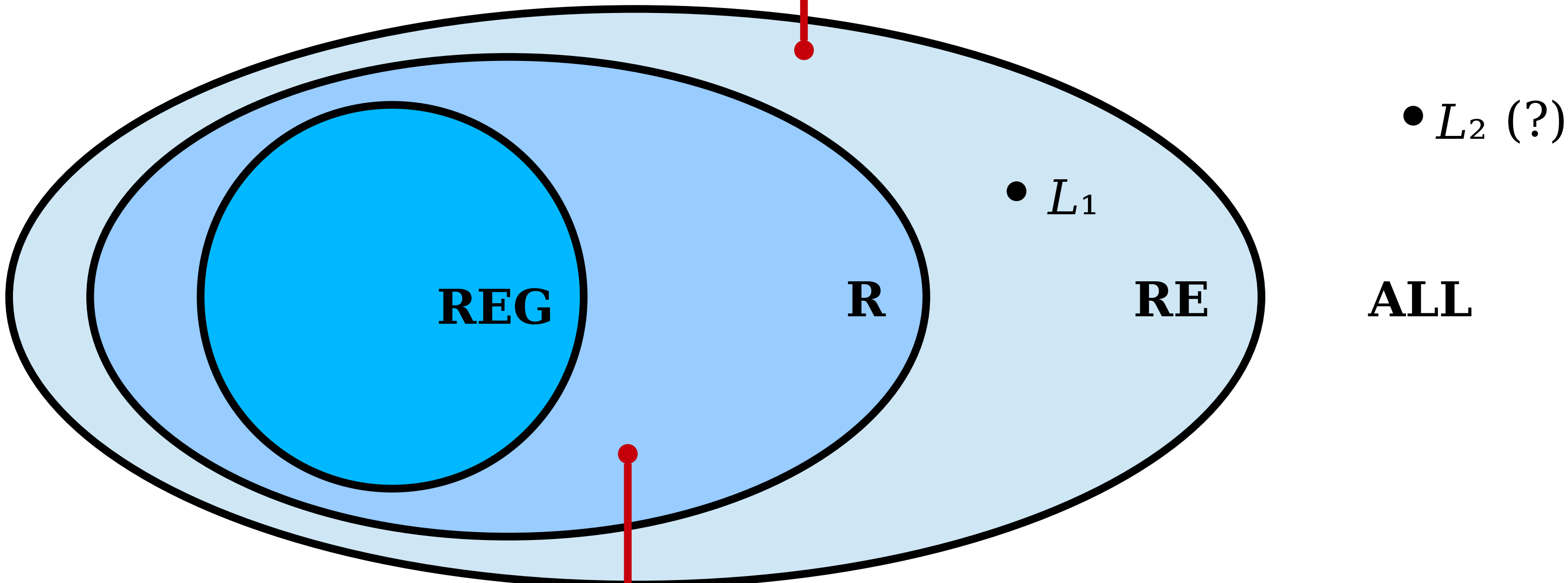
Okay! So now you've given it your best shot. Let's see where this one goes.



$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$
$L_3 = \{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N}$ and $n > 1000 \}$
$L_4 = \{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N}$ 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 $w \notin L$?

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.

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

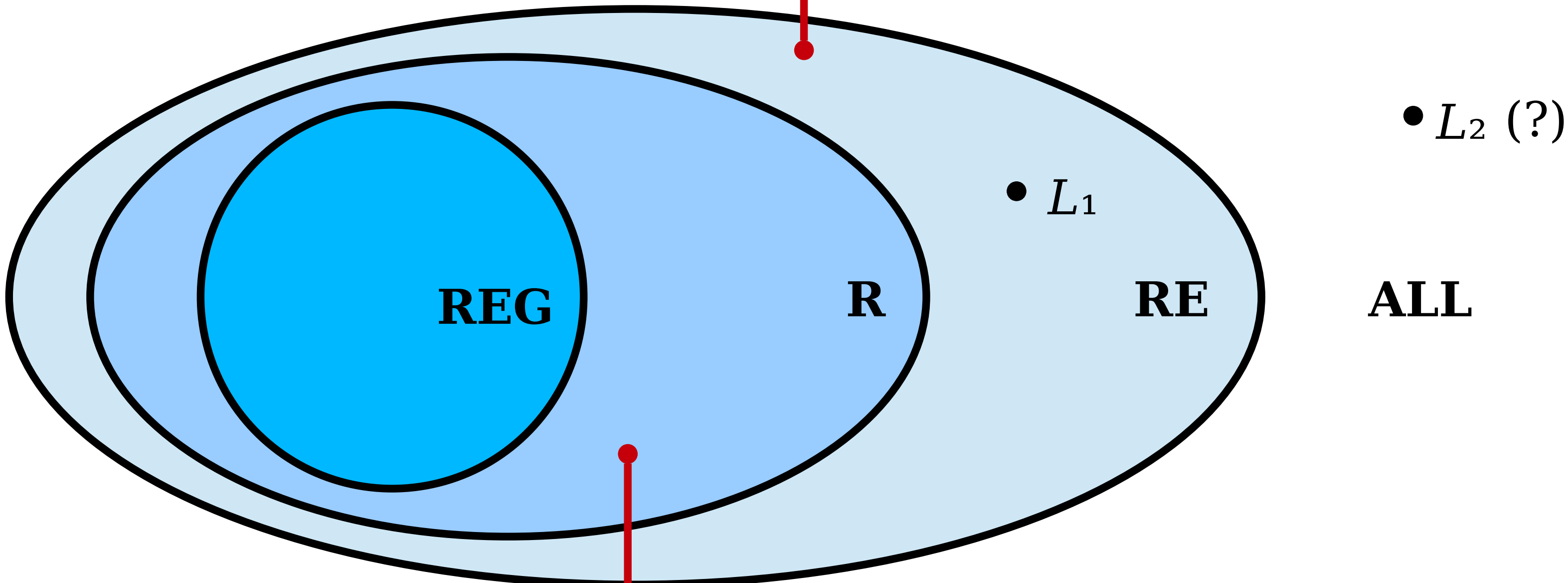
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

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

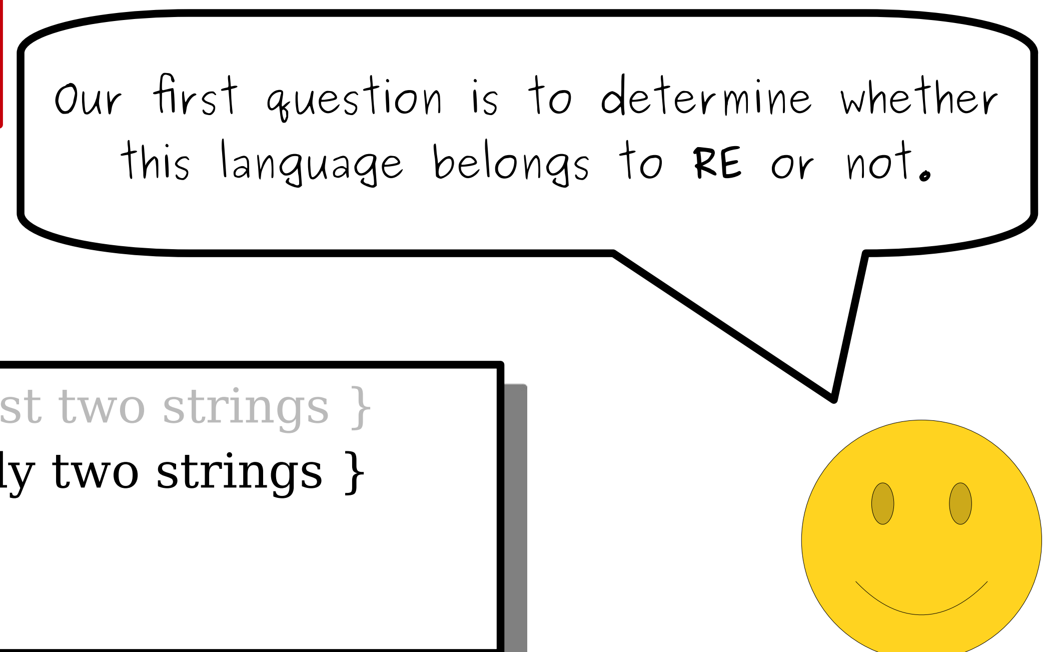
RE: Languages with Verifiers

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 $w \notin L$?



$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

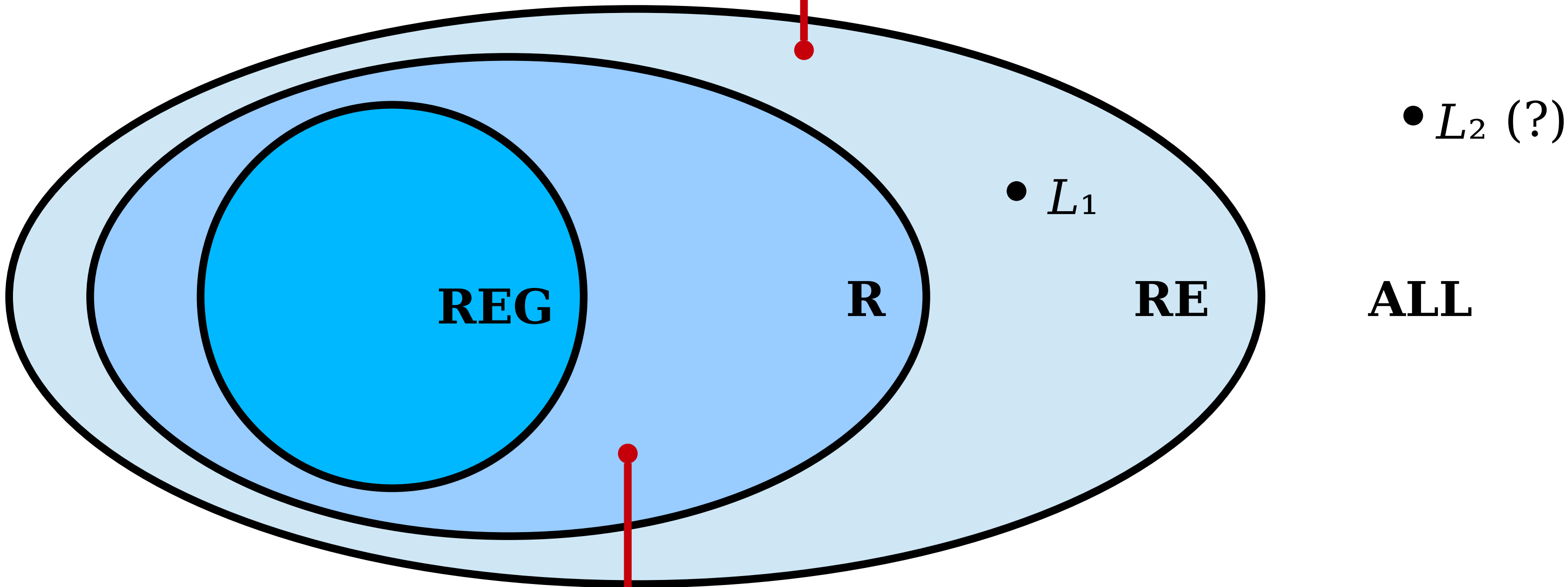
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

$L_4 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}$ 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 $w \notin L$?

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.

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

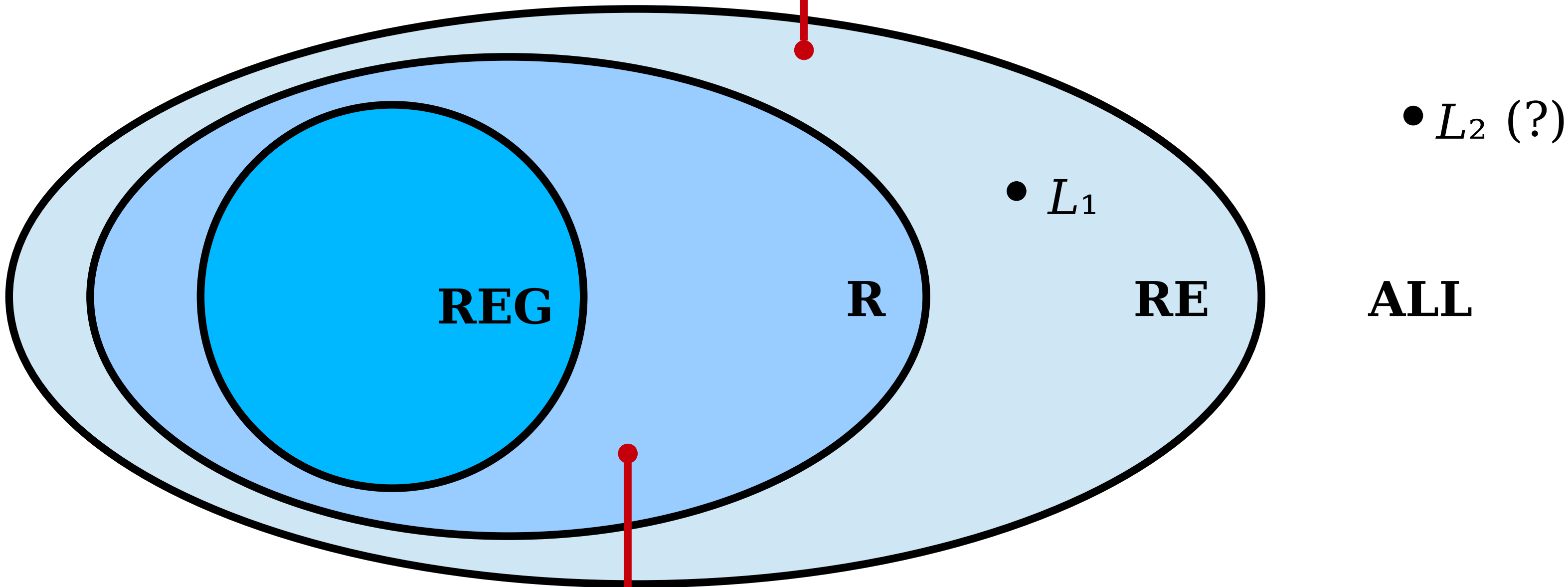
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

$L_4 = \{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N}$ 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 $w \notin L$?

This is the language of all TMs that accept exactly two strings.

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

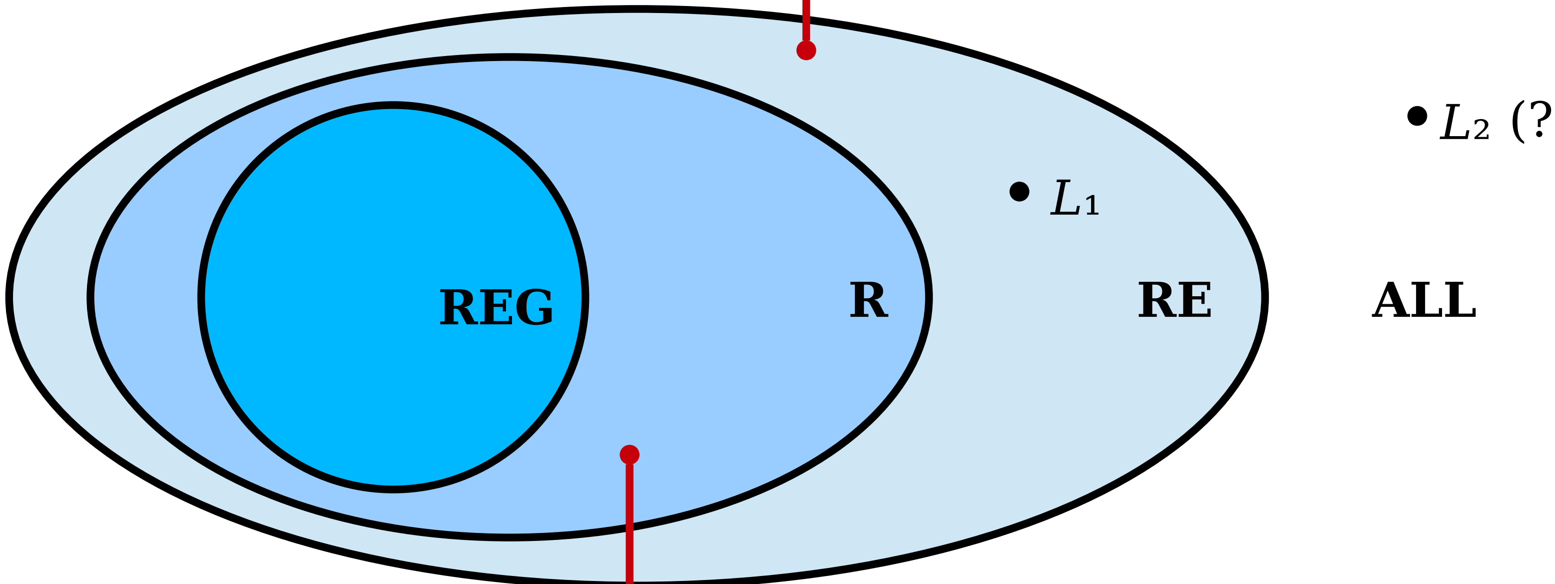
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

$L_4 = \{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N}$ 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 $w \notin L$?

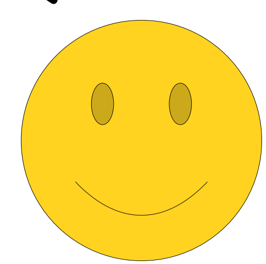
So now we ask – if had a TM and you knew for a fact that it accepted exactly two strings, could you prove it?

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM that accepts at least two strings} \}$

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM that accepts exactly two strings} \}$

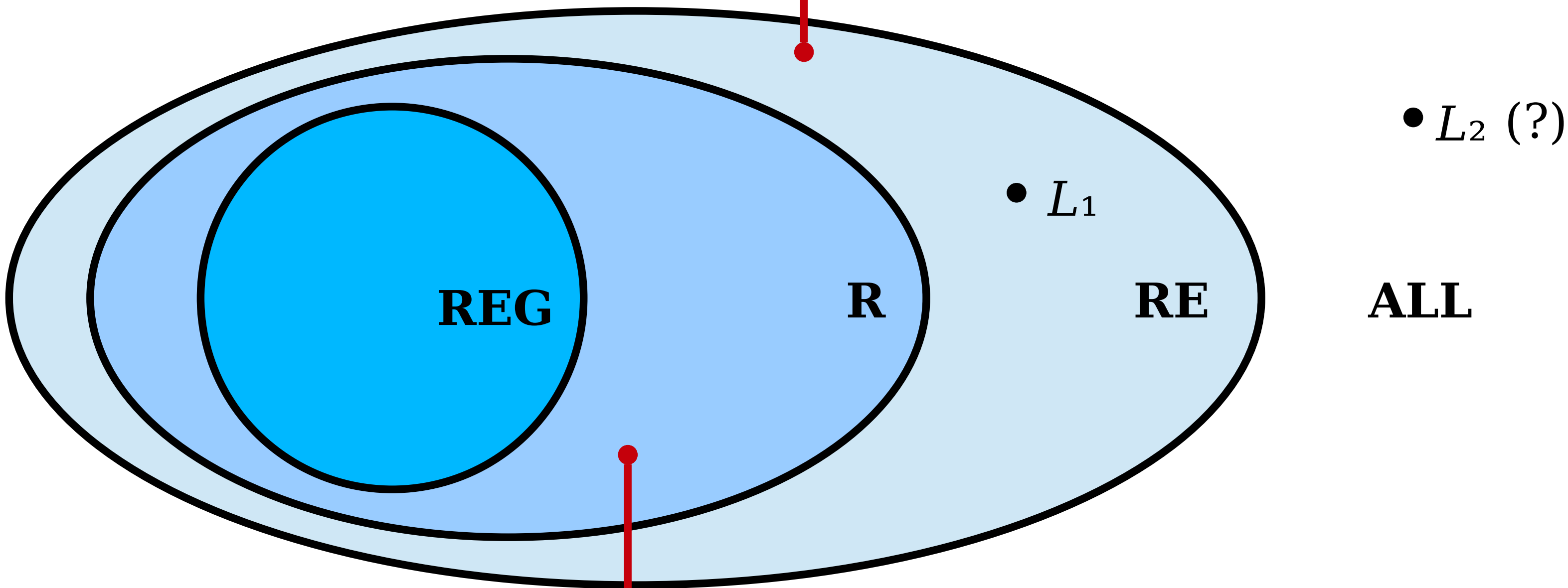
$L_3 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$

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



RE: Languages with Verifiers

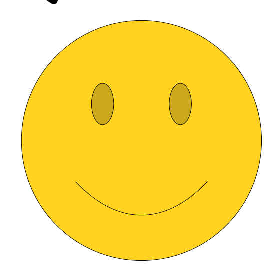
Given any string $w \in L$, could you ***prove*** that $w \in L$?



R: Languages with Deciders

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

This turns out to be a lot harder than just checking if a TM accepts at least two strings.



$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

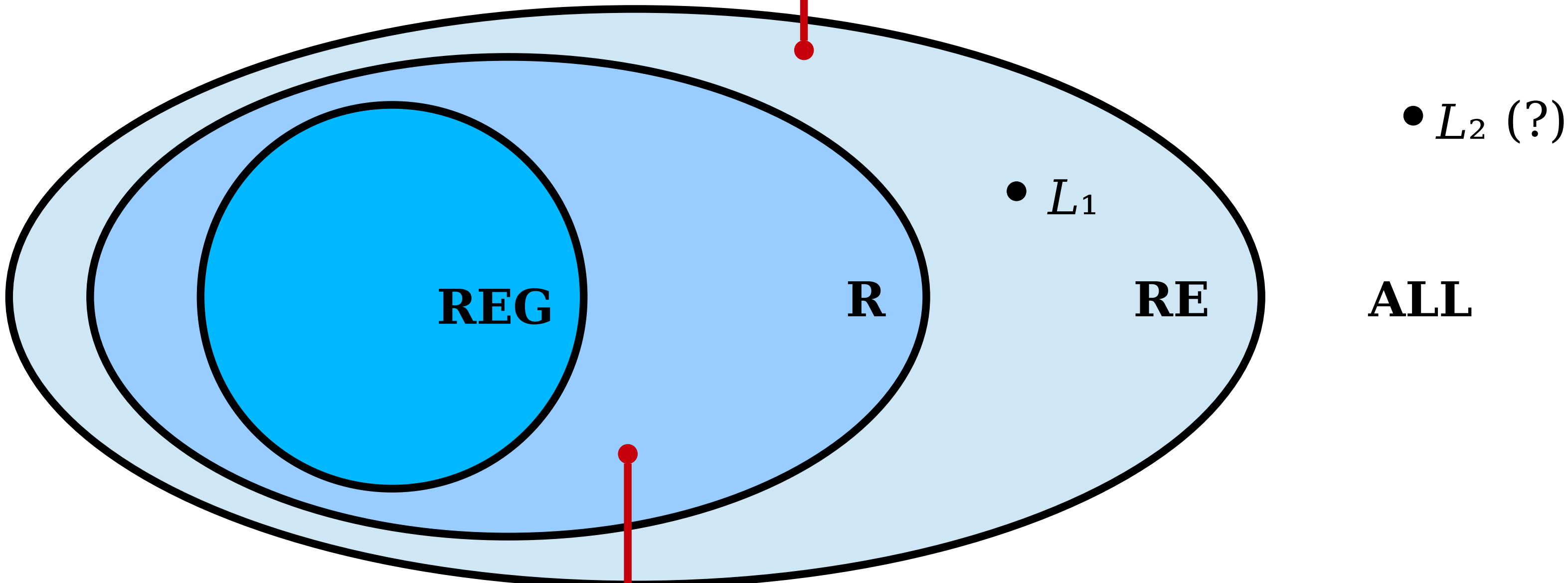
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

$L_4 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}$ 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 $w \notin L$?

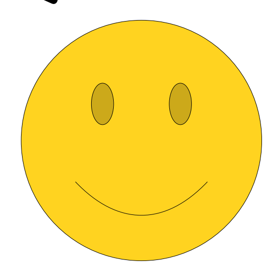
To show that a TM accepts exactly two strings, we need to show that it accepts at least two strings (that's something we can prove), but also that it doesn't accept anything else.

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

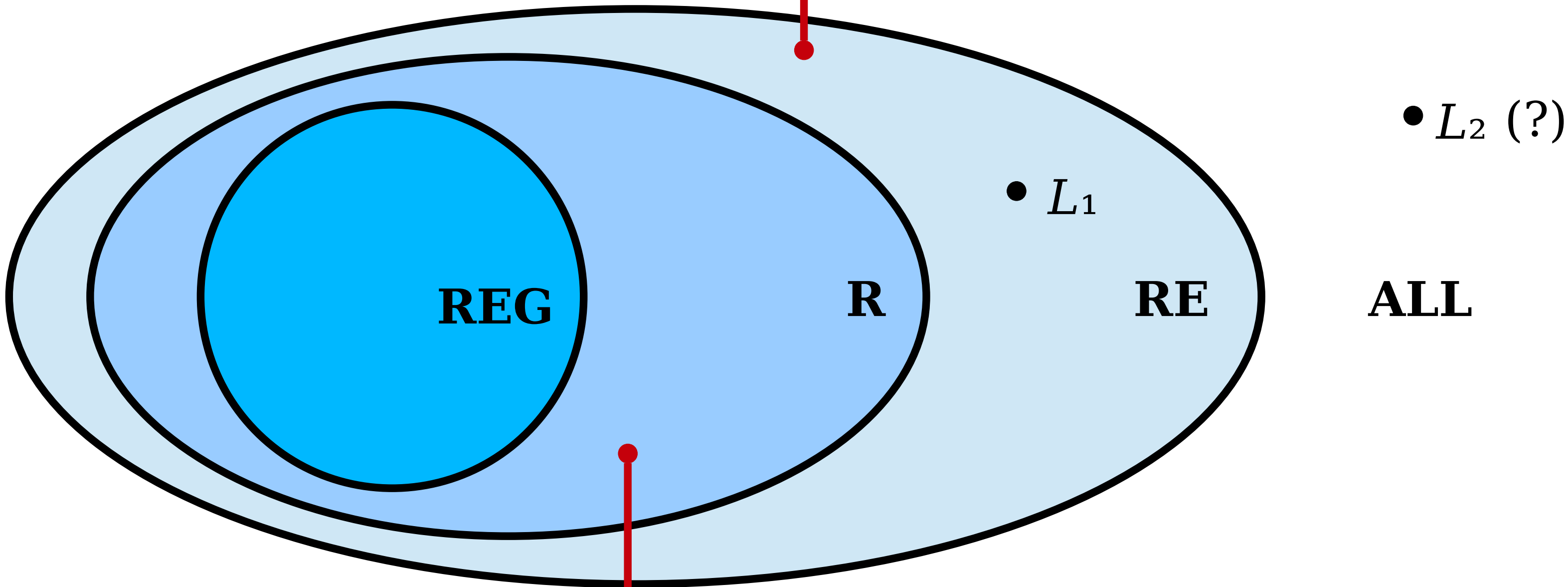
$L_3 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}$ and $n > 1000 \}$

$L_4 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}$ 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 $w \notin L$?

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.

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM that accepts at least two strings} \}$

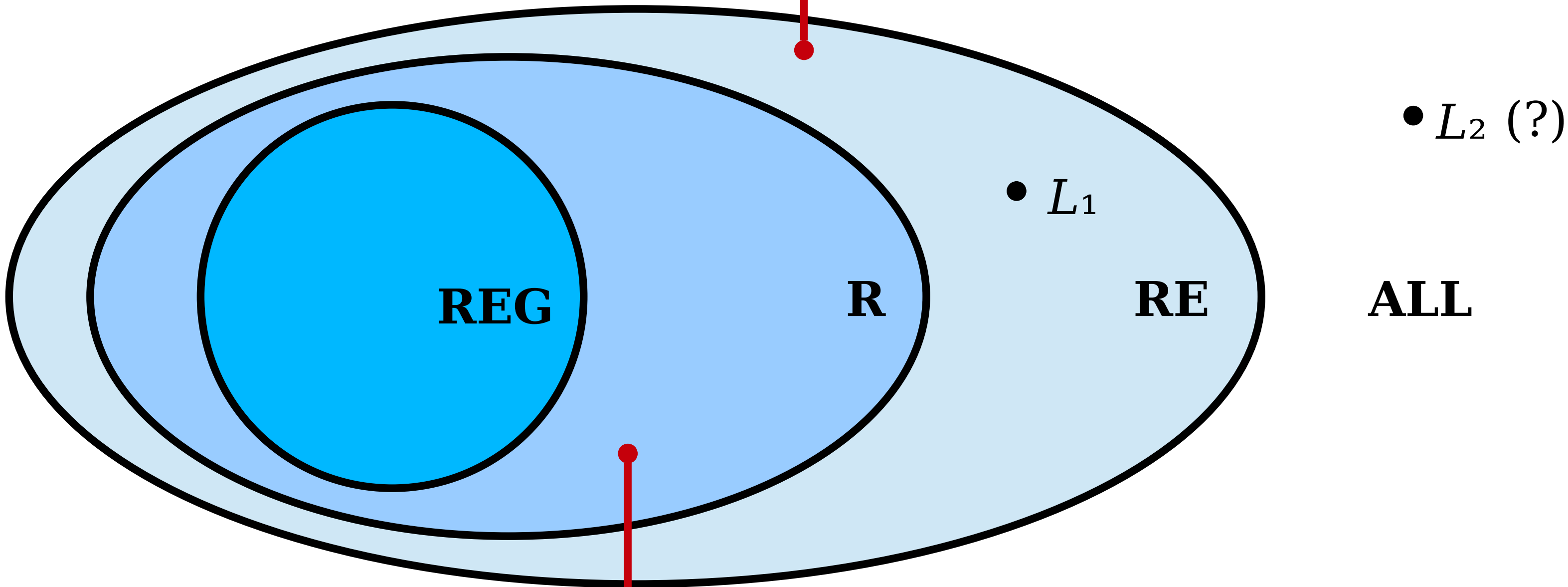
$L_2 = \{ \langle M \rangle \mid M \text{ is a TM that accepts exactly two strings} \}$

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

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

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 $w \notin L$?

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.

L_1 = { ⟨M⟩ | M is a TM that accepts at least two strings }

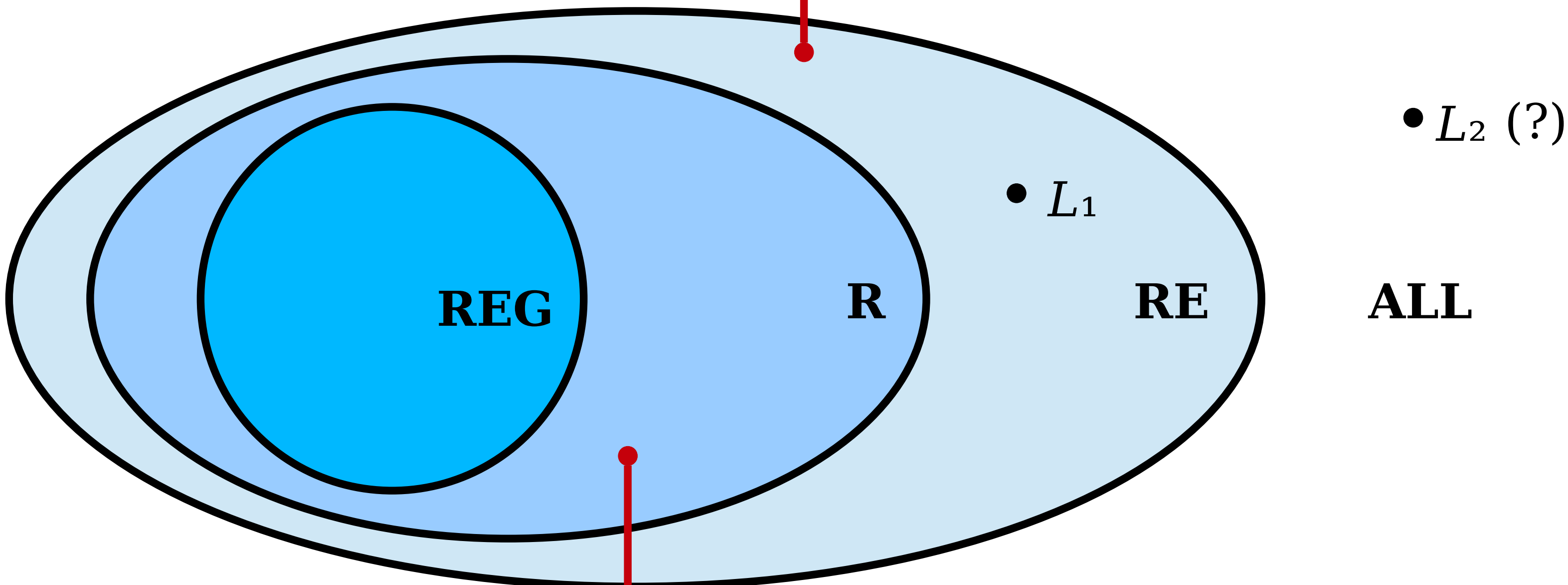
L_2 = { ⟨M⟩ | M is a TM that accepts exactly two strings }

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

L_4 = { $\mathtt{a}^n\mathtt{b}^n$ | $n \in \mathbb{N}$ 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 $w \notin L$?

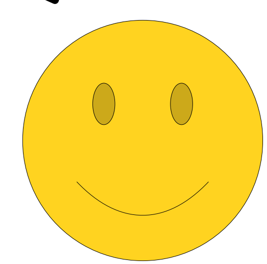
So at least, intuitively, this doesn't seem like it's going to be possible to do. Even if we know that TM accepts exactly two strings, it's unclear how we'd prove that to someone.

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

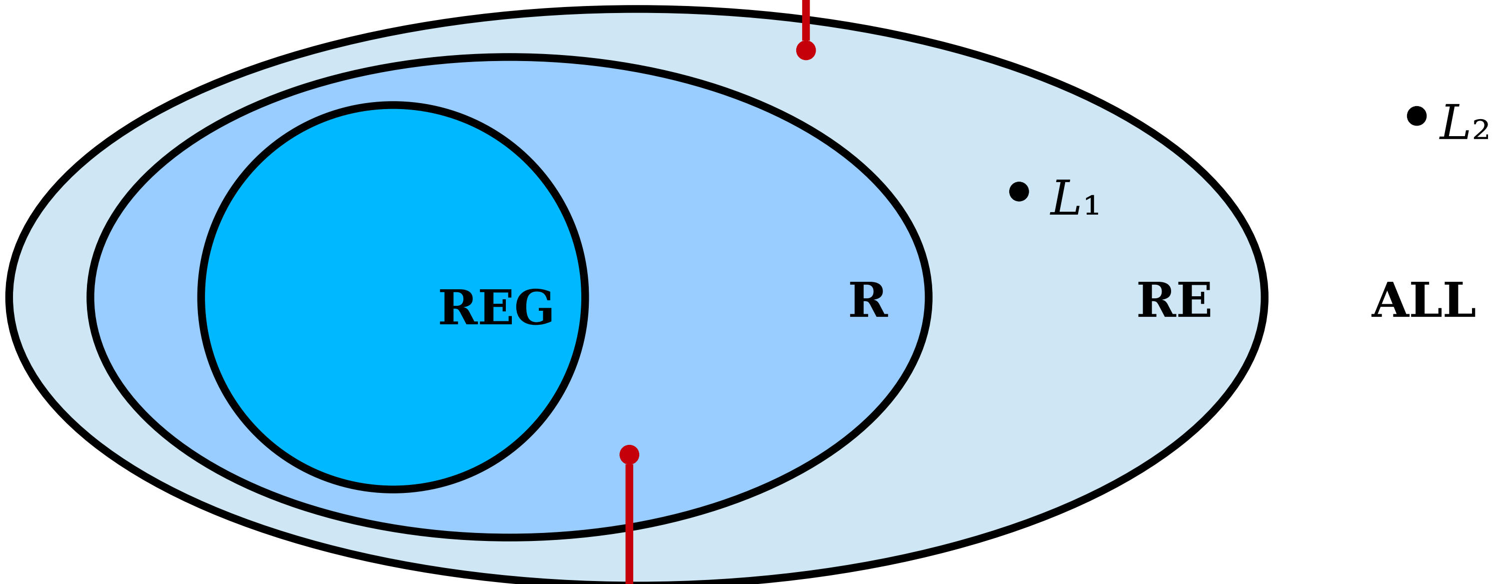
$L_3 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}$ and $n > 1000 \}$

$L_4 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}$ 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 $w \notin L$?

This gives us some justification to guess that this language is probably not going to be in **RE**.

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

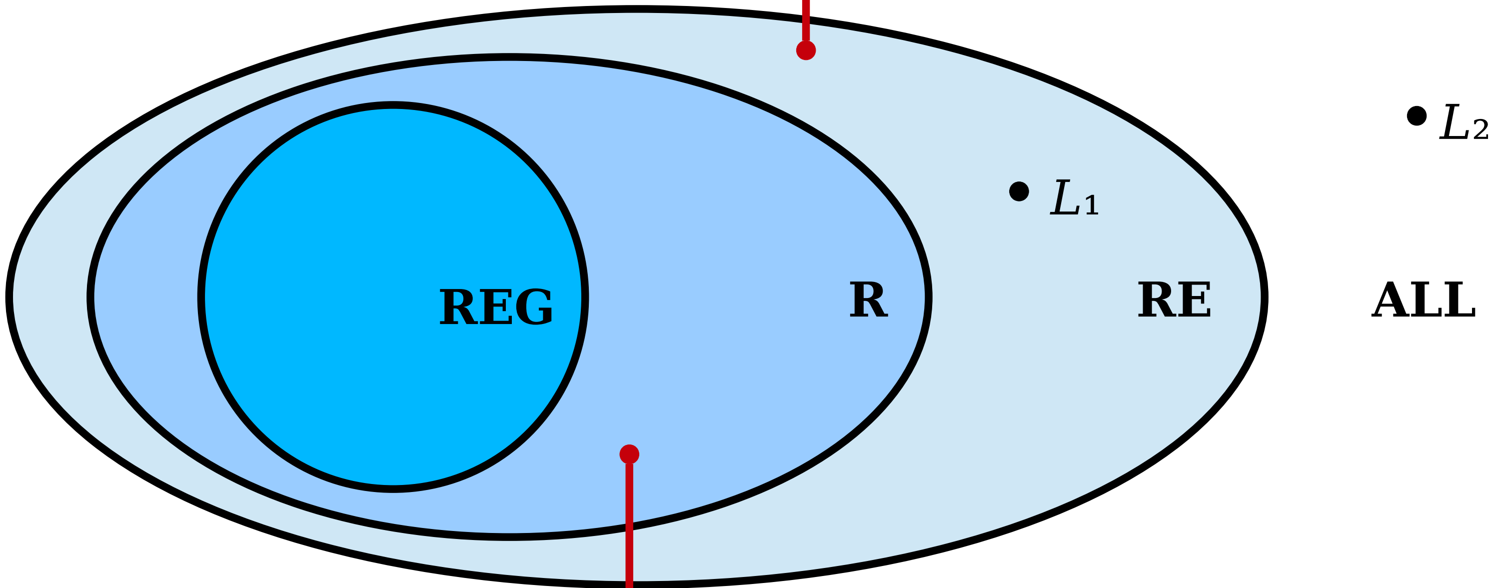
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

$L_4 = \{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N}$ 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

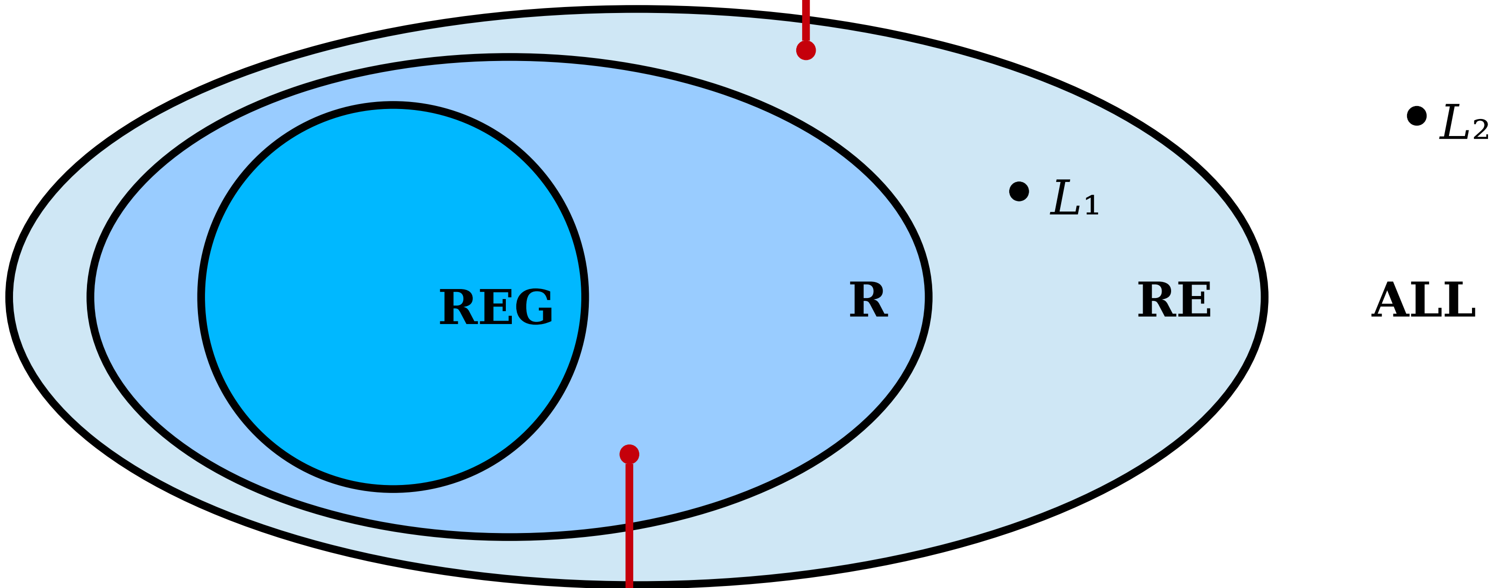
In addition to the **RE** requirements, given any string $w \notin L$, could you ***prove*** that $w \notin L$?

So there you have it – this language is not even in RE.

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$
$L_3 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}$ and $n > 1000 \}$
$L_4 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}$ 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 $w \notin L$?

That might seem pretty surprising, given how similar this language looks to L_1.

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

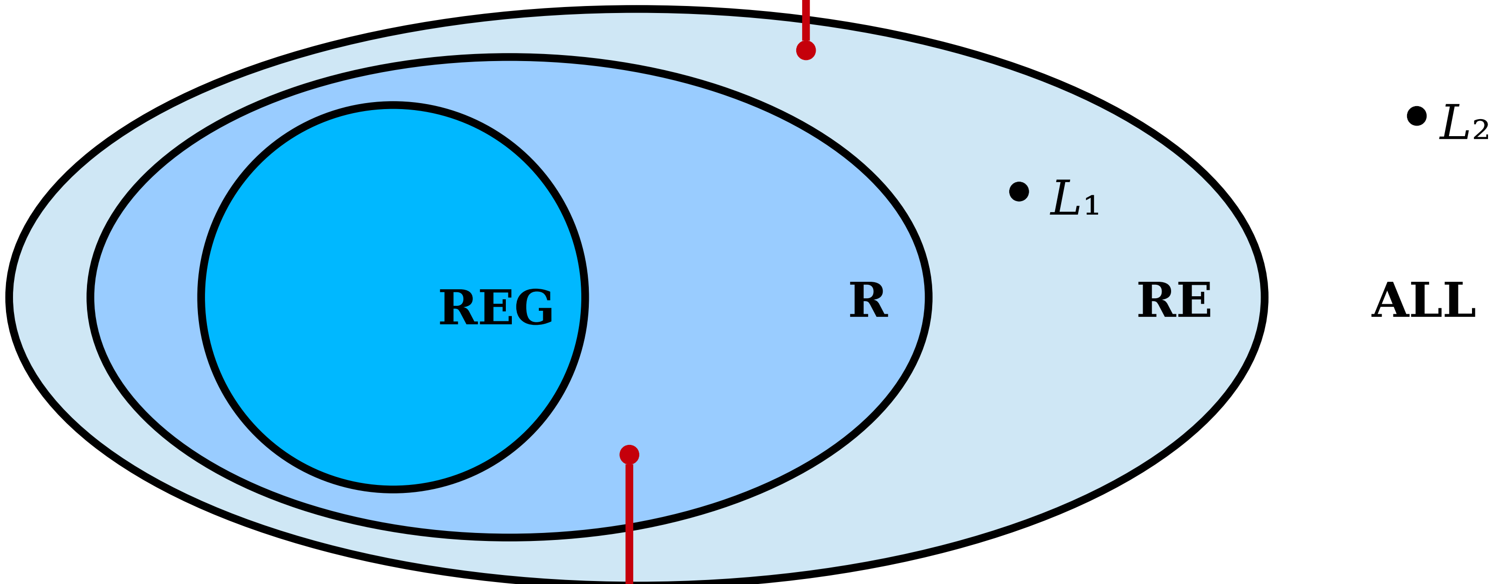
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

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

RE: Languages with Verifiers

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

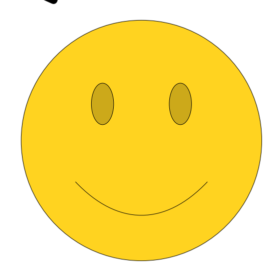


R: Languages with Deciders

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

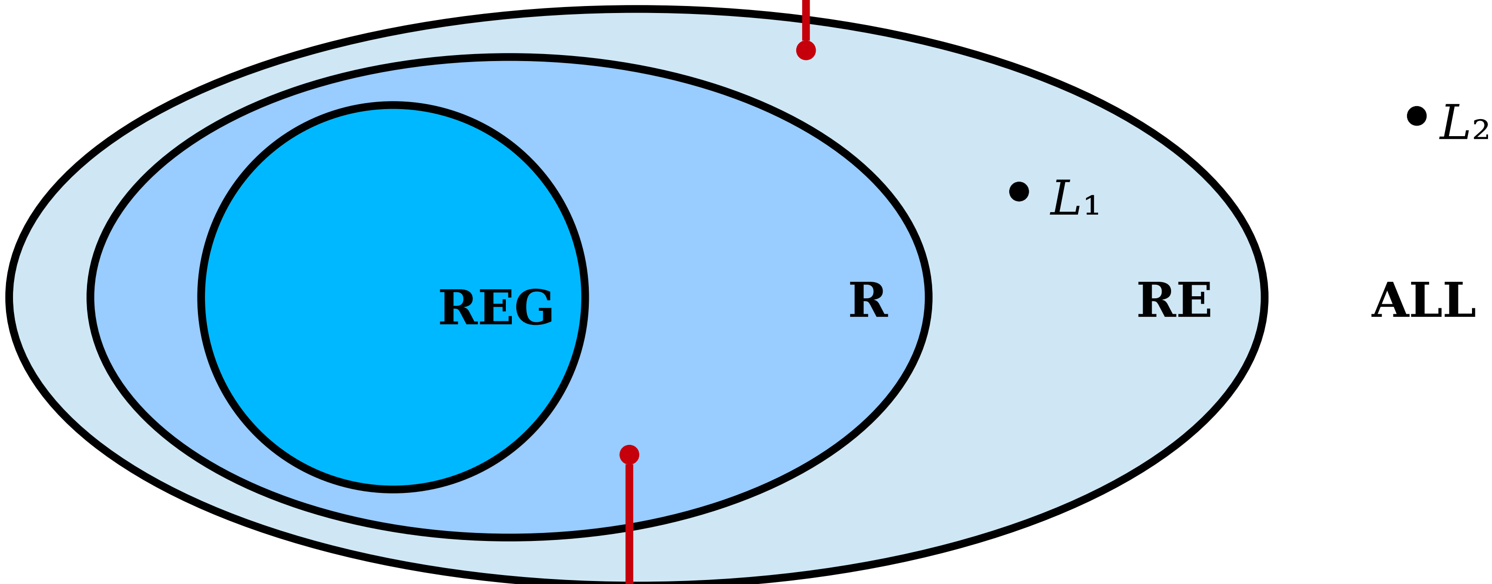
I chose this particular example because it highlights a key point when thinking about languages: don't try to place a language in the diagram just based on its description.

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$
$L_3 = \{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N}$ and $n > 1000 \}$
$L_4 = \{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N}$ 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 $w \notin L$?

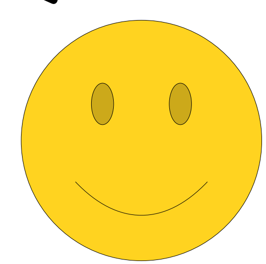
To figure out where something goes, you need to think about in terms of provability. Ultimately, it's this – rather than the way it's written – that makes things hard.

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

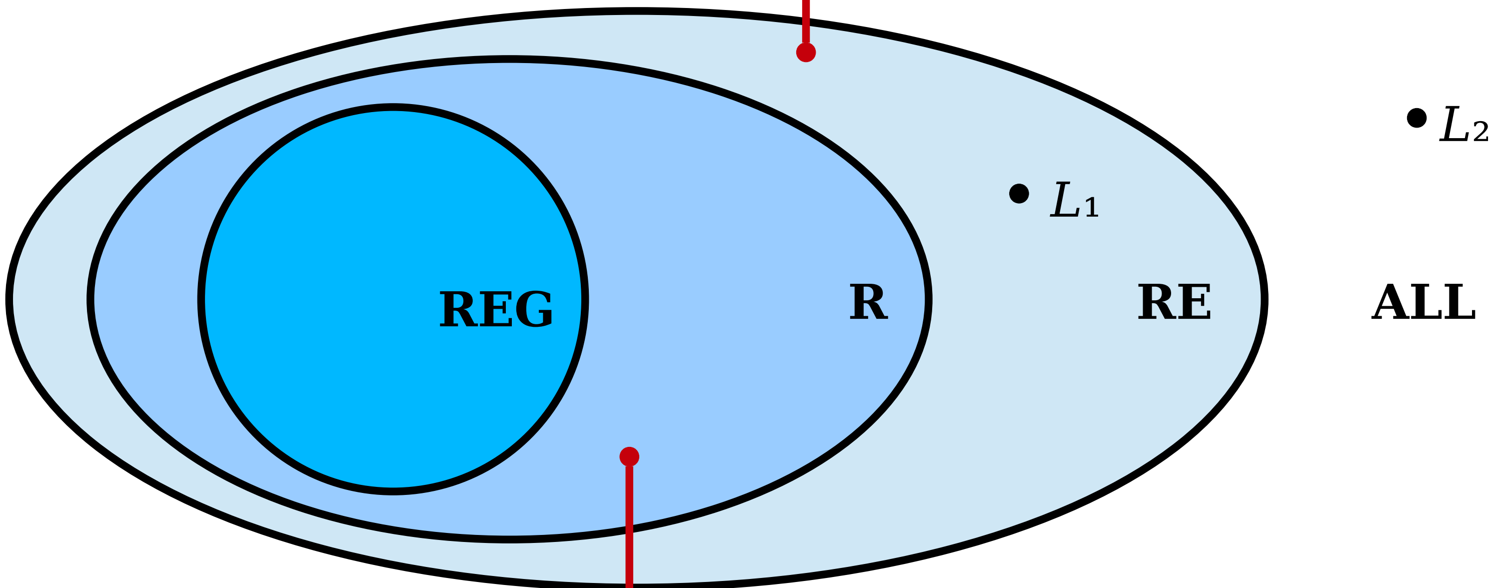
$L_3 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}$ and $n > 1000 \}$

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



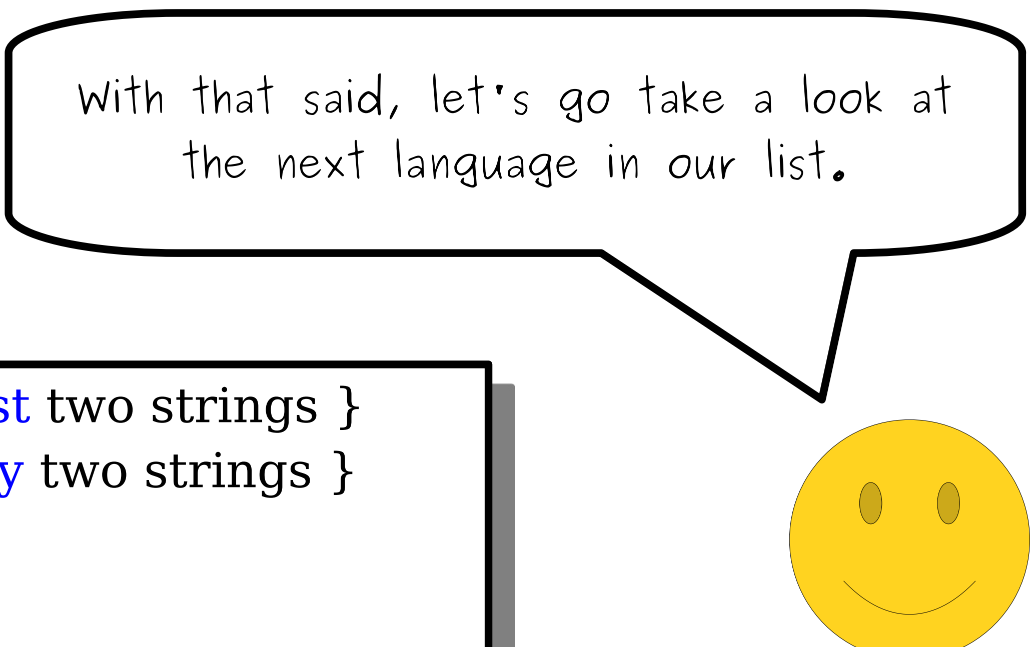
RE: Languages with Verifiers

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 $w \notin L$?



$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

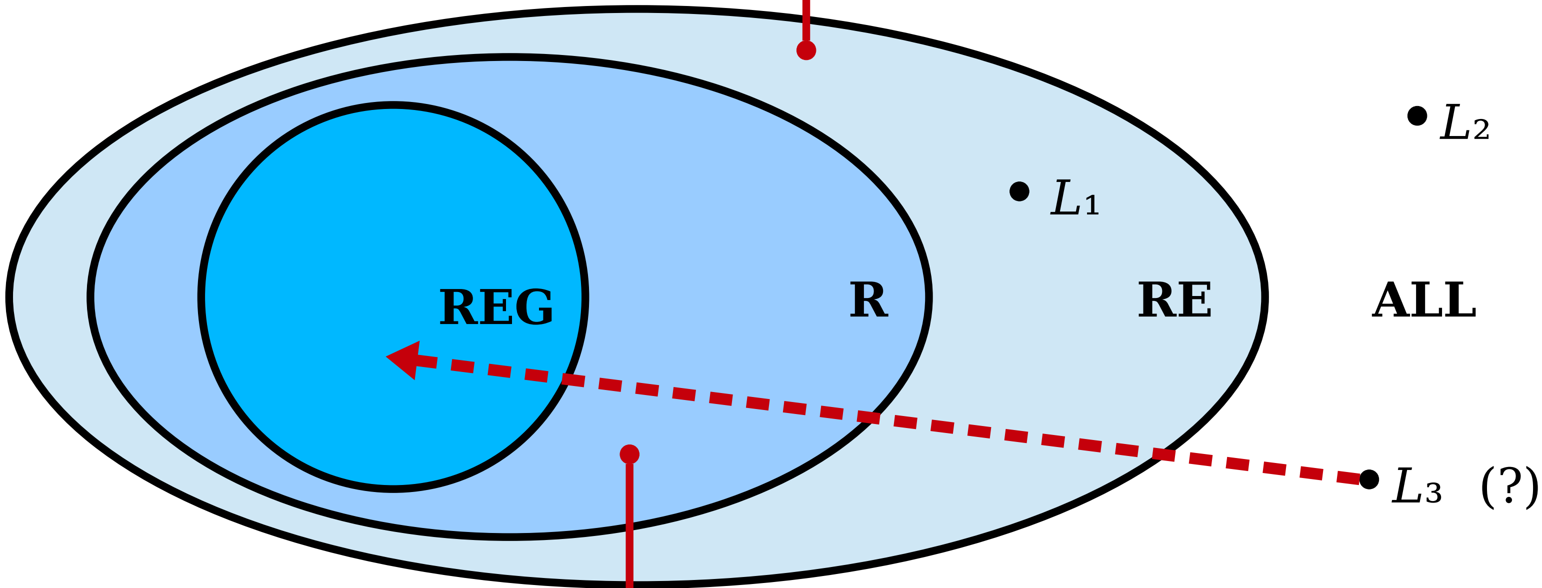
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

$L_4 = \{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N}$ 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 $w \notin L$?

As before, we'll start by placing it outside of **RE** and try to think about pushing it as far down as possible.

L_1 = { ⟨M⟩ | M is a TM that accepts at least two strings }

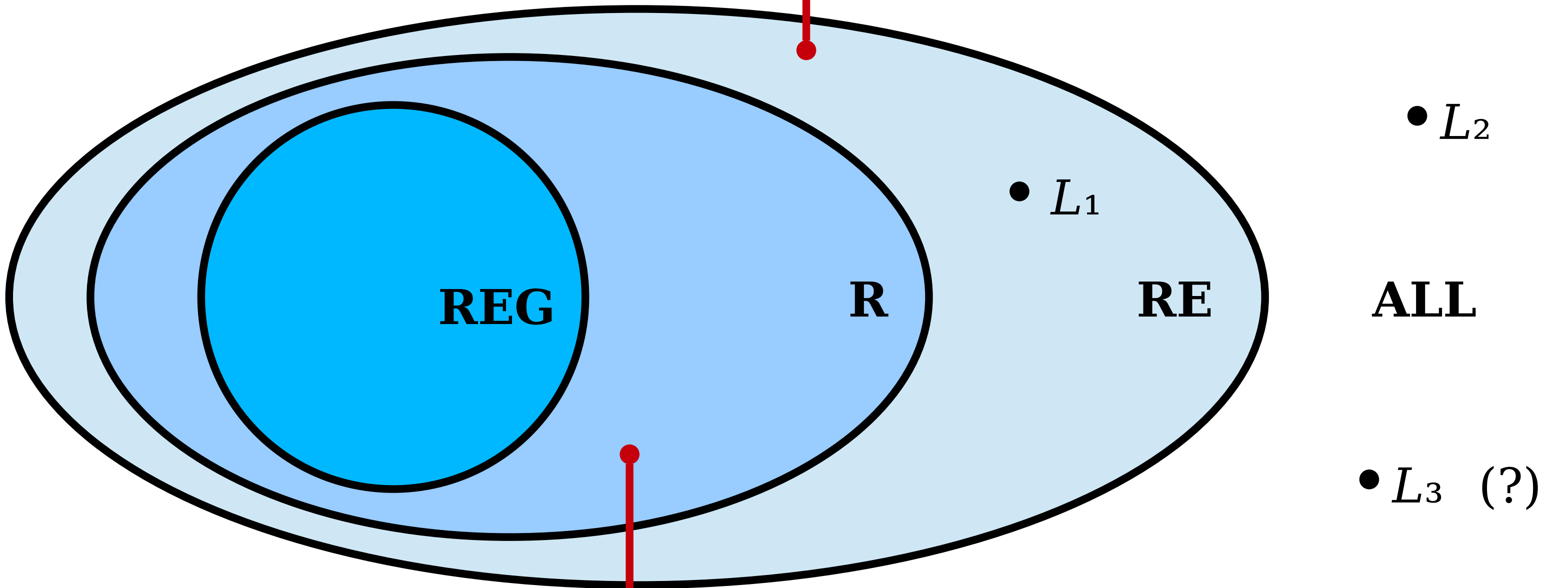
L_2 = { ⟨M⟩ | M is a TM that accepts exactly two strings }

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

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

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 $w \notin L$?

As before, we first ask whether this language happens to be in **RE**.

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

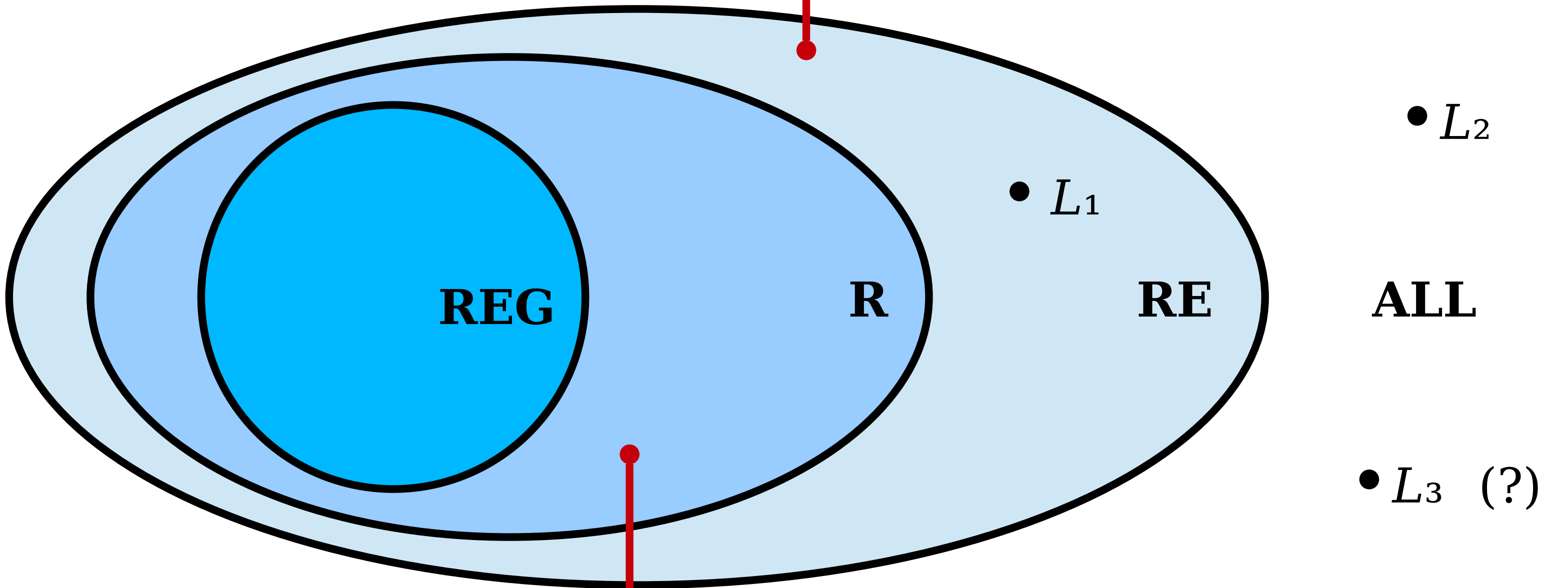
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

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

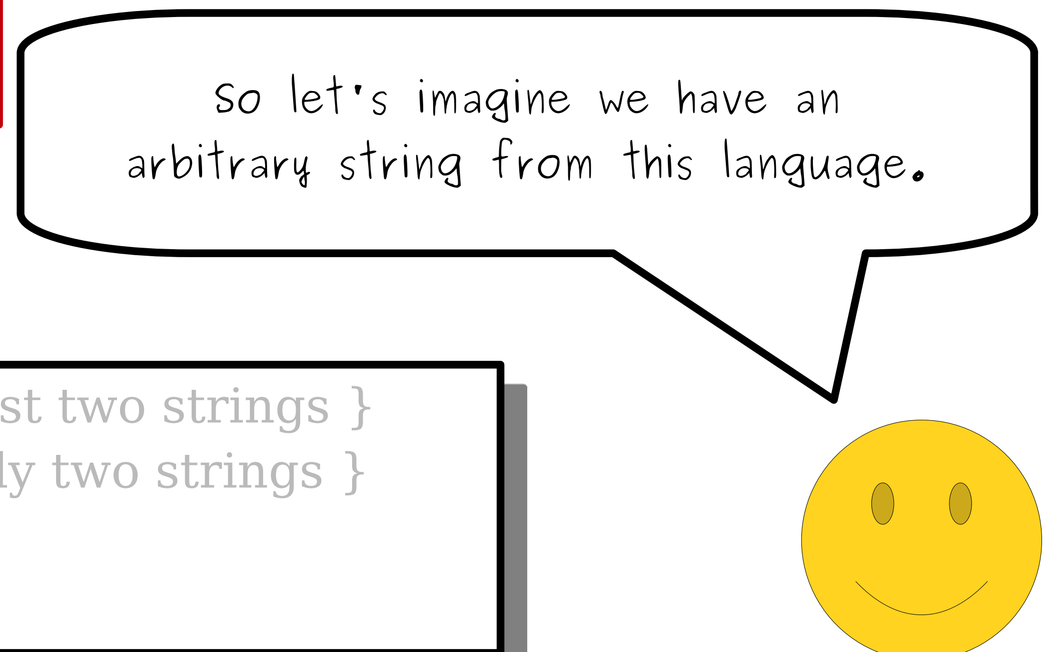
RE: Languages with Verifiers

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 $w \notin L$?



$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

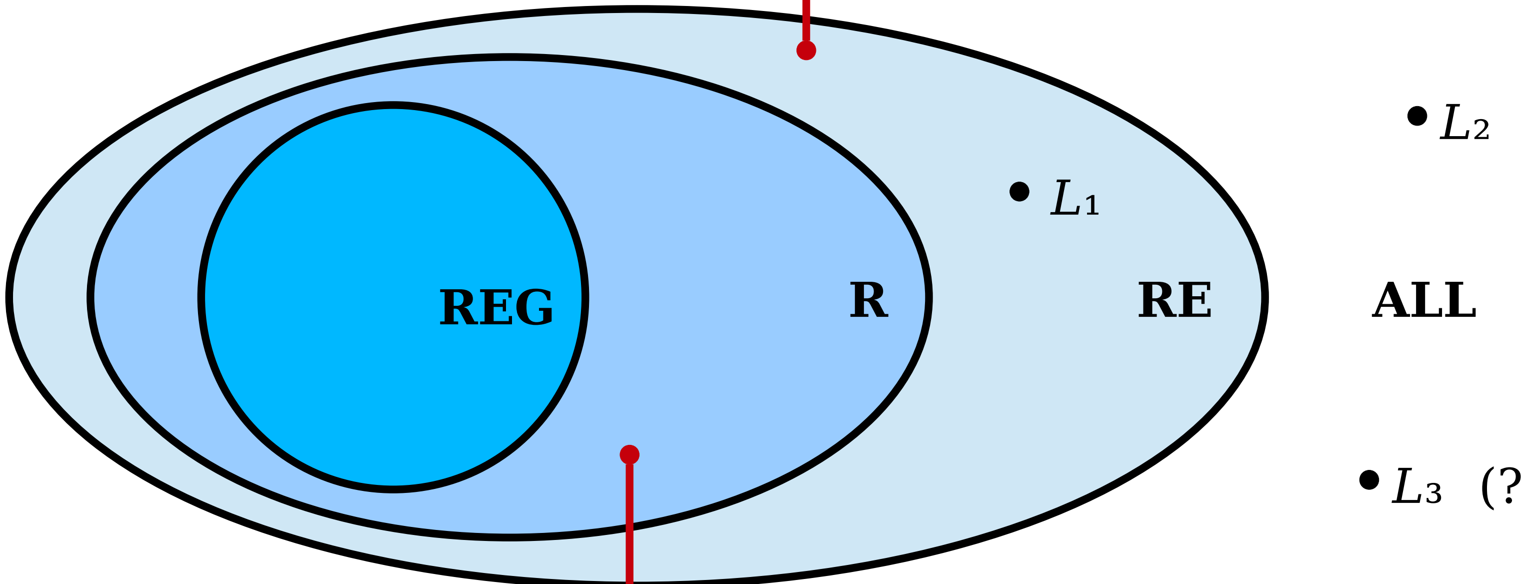
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

$L_4 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}$ 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 $w \notin L$?

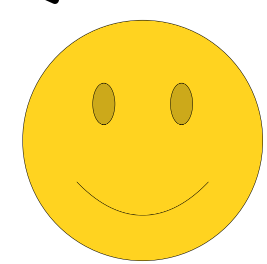
That means that we have a string of the form $a^n b^n$ with at least 2,002 characters in it (at least 1,001 a's and at least 1,001 b's.)

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

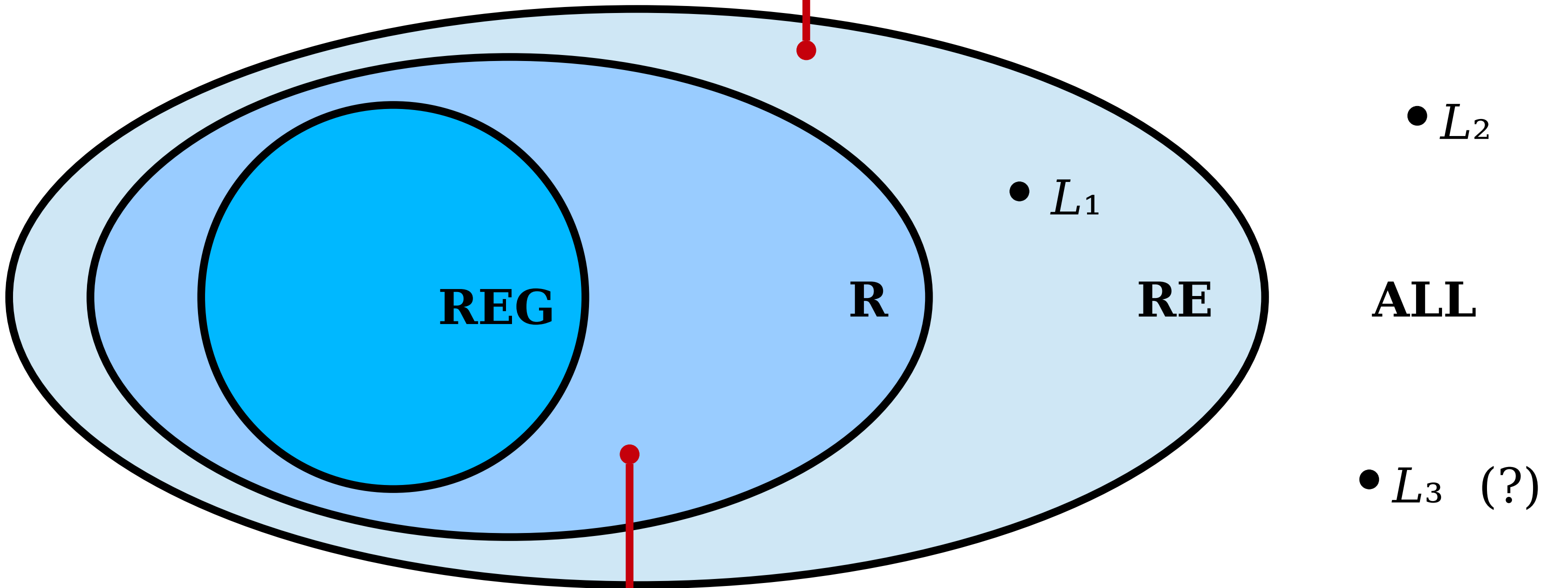
$L_3 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}$ and $n > 1000 \}$

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



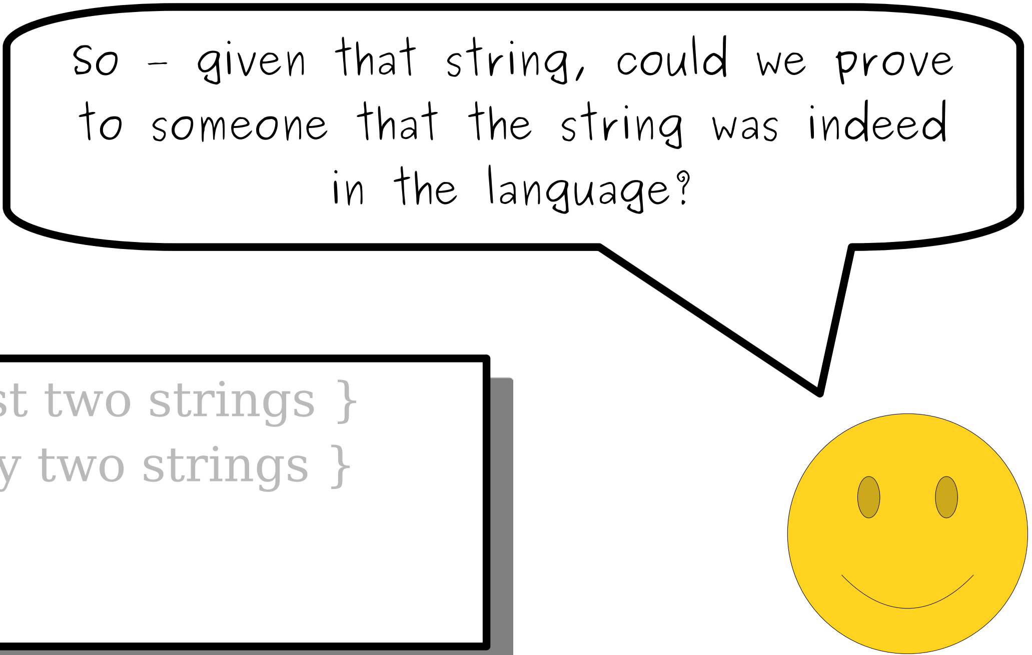
RE: Languages with Verifiers

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 $w \notin L$?



$L_1 = \{ \langle M \rangle \mid M \text{ is a TM that accepts at least two strings} \}$

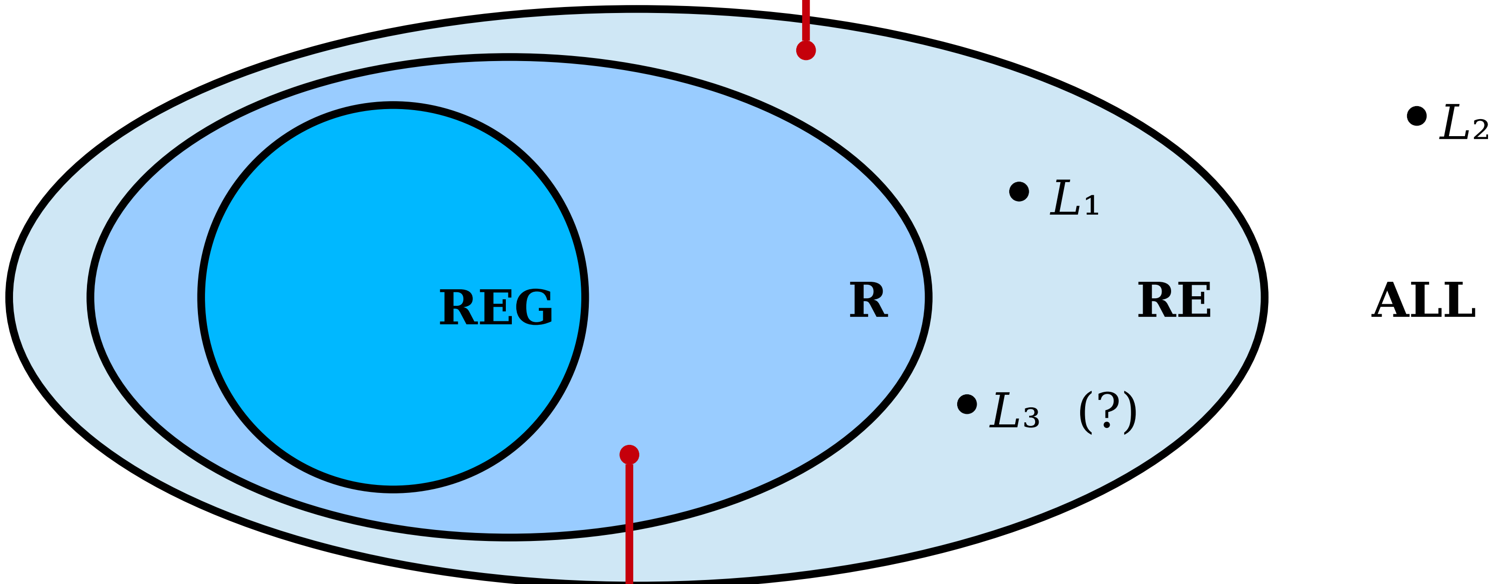
$L_2 = \{ \langle M \rangle \mid M \text{ is a TM that accepts exactly two strings} \}$

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

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

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 $w \notin L$?

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 = { ⟨M⟩ | M is a TM that accepts at least two strings }

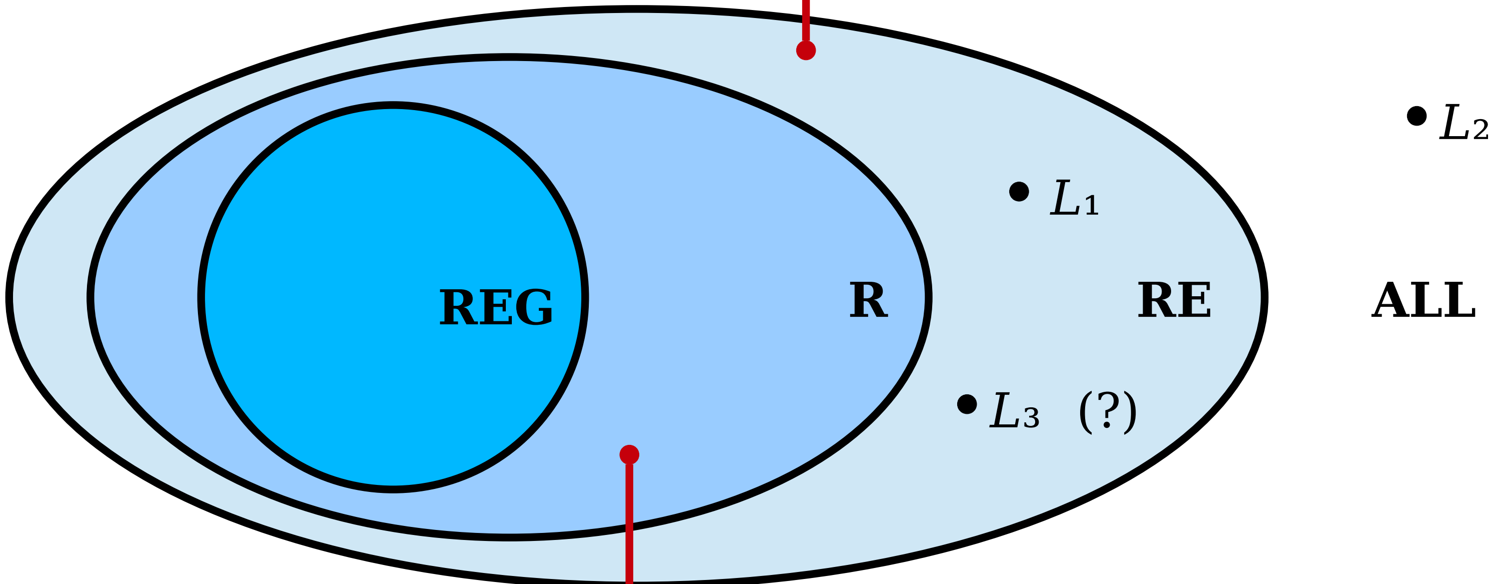
L_2 = { ⟨M⟩ | M is a TM that accepts exactly two strings }

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

L_4 = { $\mathbf{a}^n\mathbf{b}^n$ | $n \in \mathbb{N}$ 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 $w \notin L$?

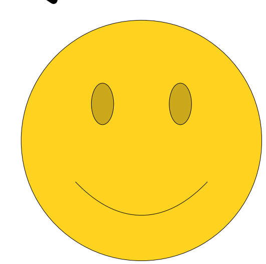
Next, let's ask the follow-up question to see if L_3 is in **R**. If we had a string not in the language, could we prove it?

L_1 = { ⟨M⟩ | M is a TM that accepts at least two strings }

L_2 = { ⟨M⟩ | M is a TM that accepts exactly two strings }

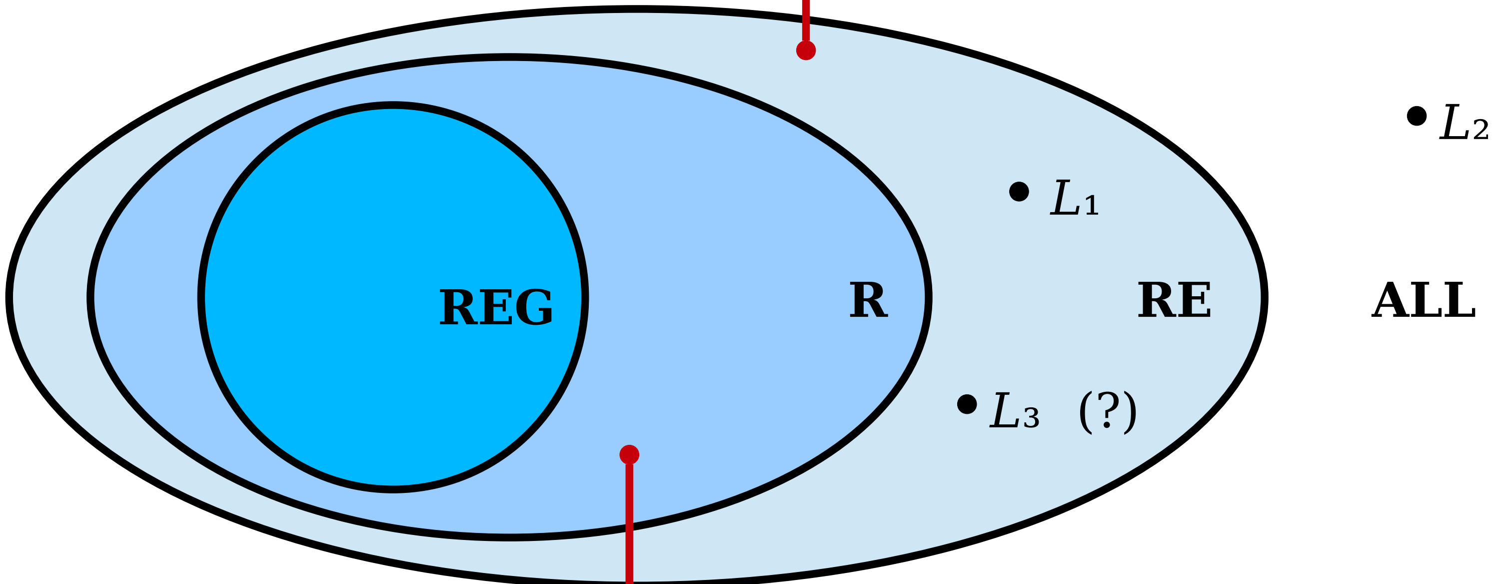
$L_3 = \{ \texttt{a}^n\texttt{b}^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$

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



RE: Languages with Verifiers

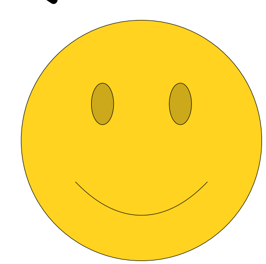
Given any string $w \in L$, could you ***prove*** that $w \in L$?



R: Languages with Deciders

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

There are a lot of cases to check if the string ends up not being in the language.



$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

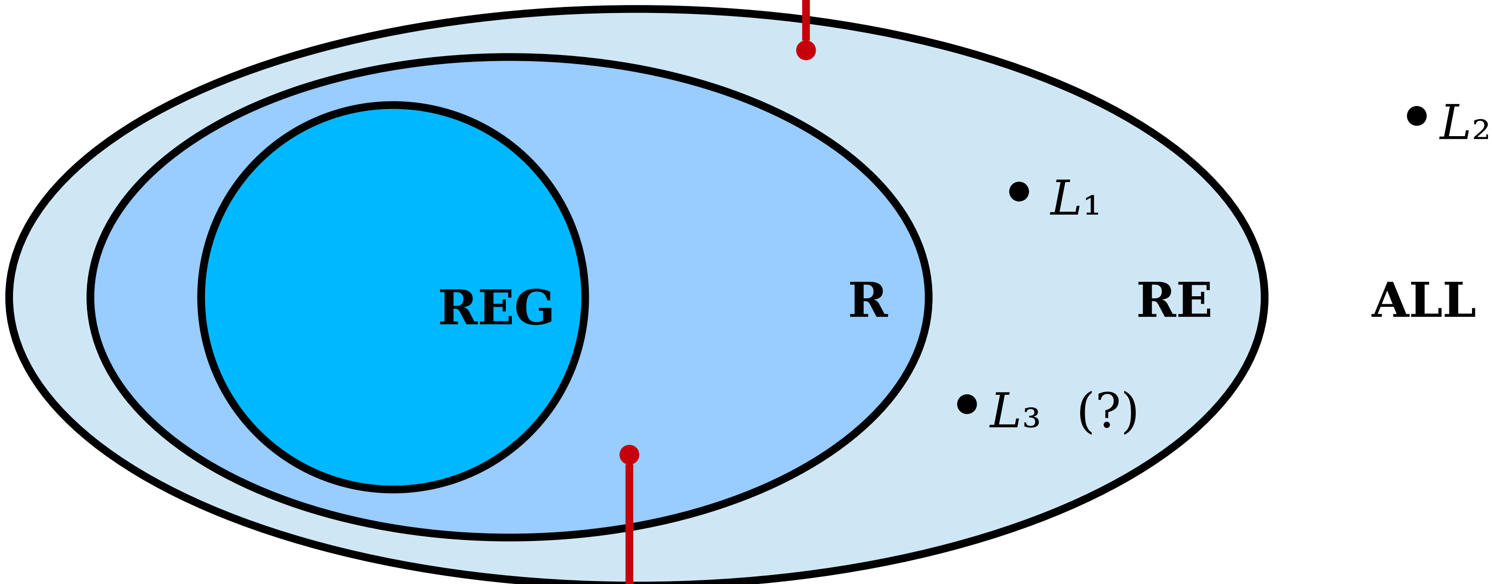
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

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

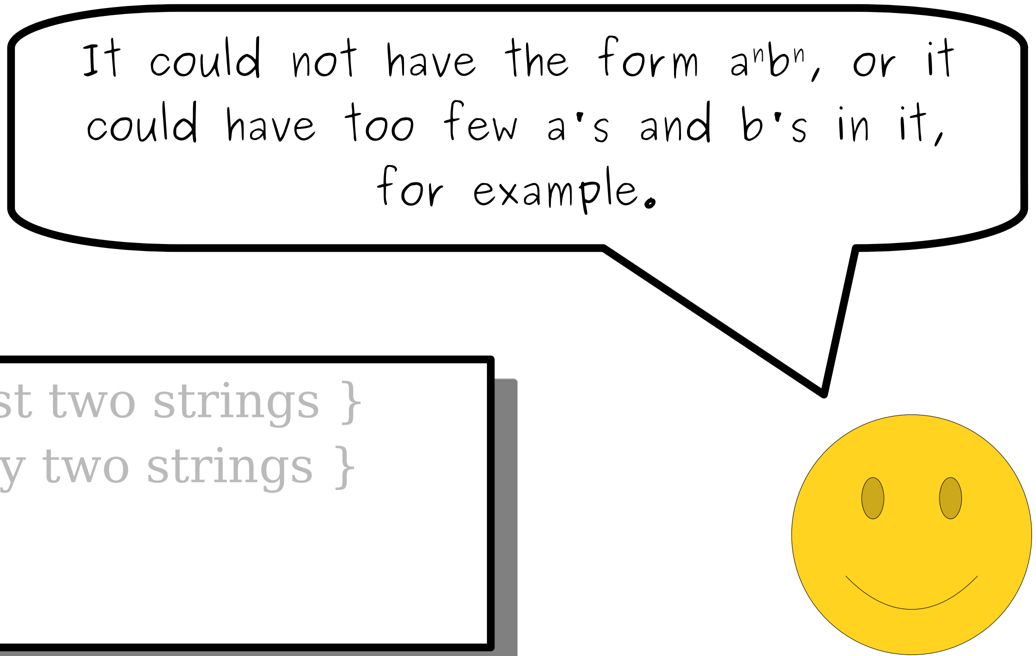
RE: Languages with Verifiers

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 $w \notin L$?



L_1 = { ⟨M⟩ | M is a TM that accepts at least two strings }

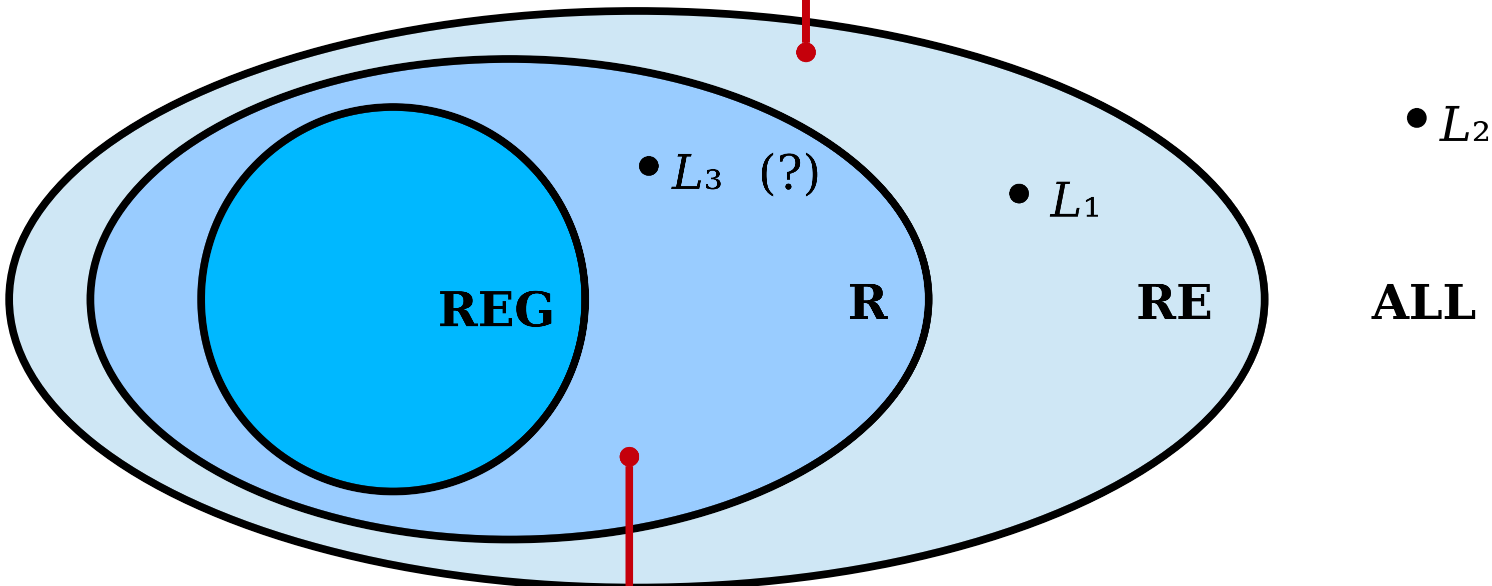
L_2 = { ⟨M⟩ | M is a TM that accepts exactly two strings }

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

L_4 = { $\mathtt{a}^n\mathtt{b}^n$ | $n \in \mathbb{N}$ 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 $w \notin L$?

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.

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

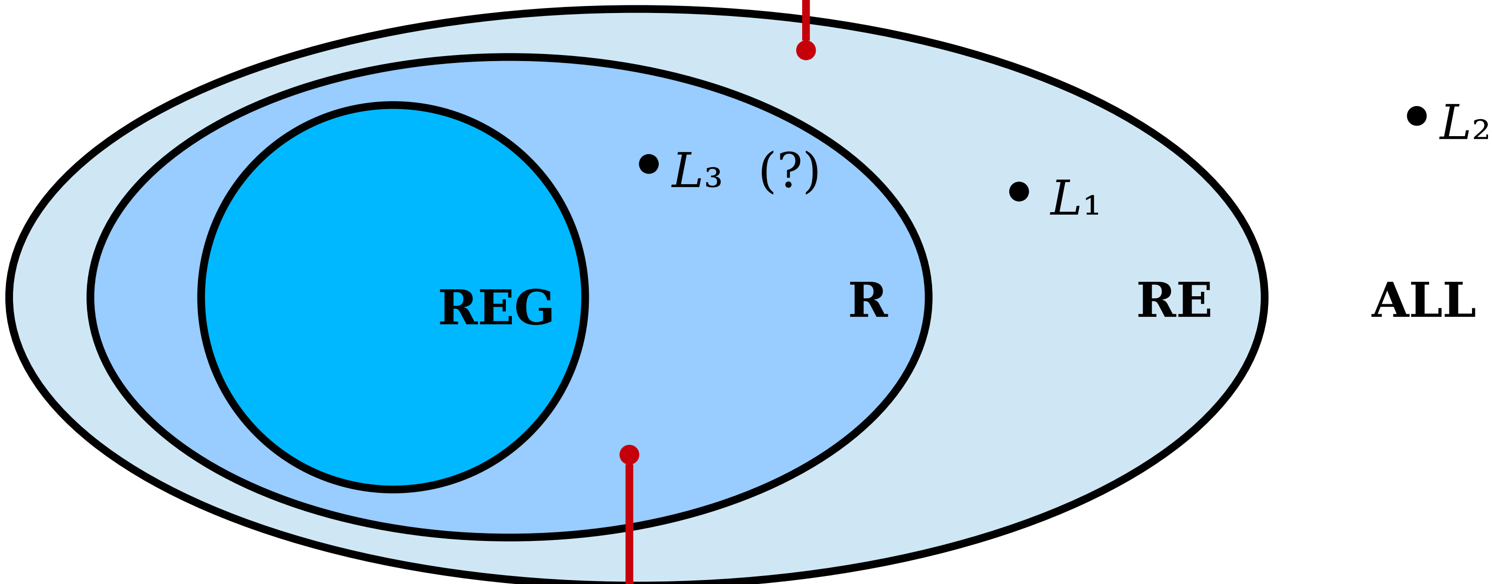
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

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

RE: Languages with Verifiers

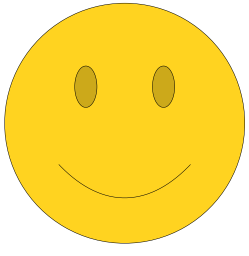
Given any string $w \in L$, could you ***prove*** that $w \in L$?



R: Languages with Deciders

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

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.



L_1 = { ⟨M⟩ | M is a TM that accepts at least two strings }

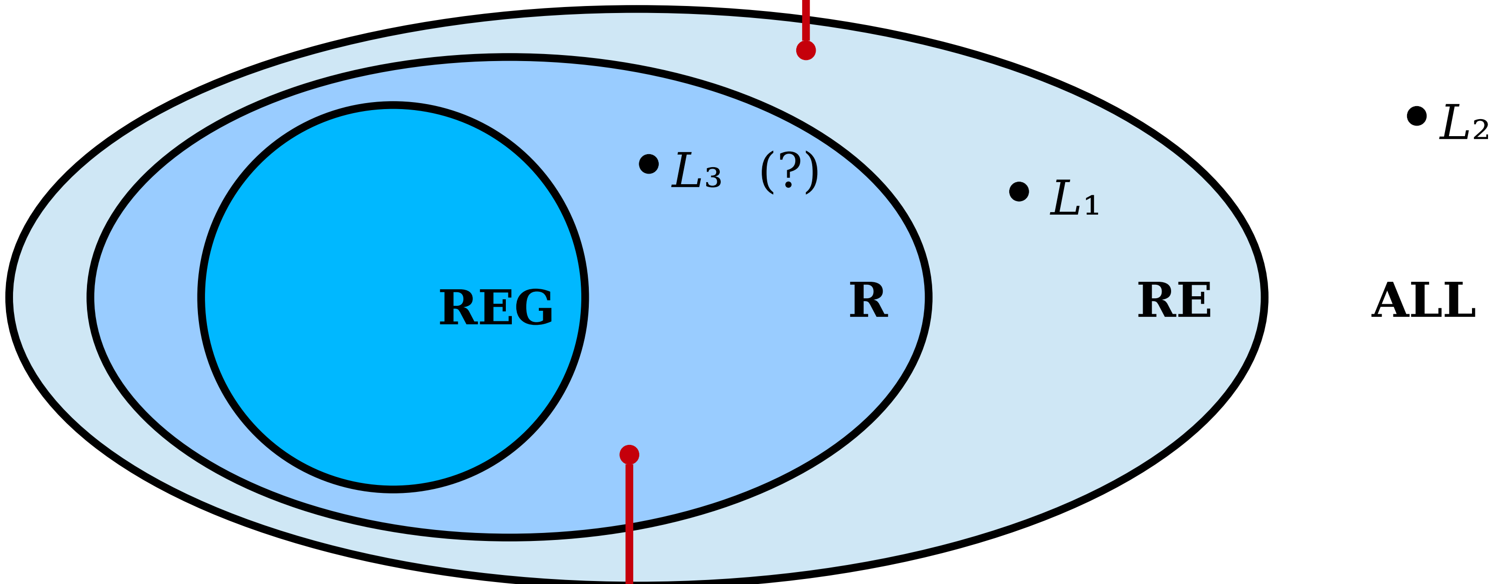
L_2 = { ⟨M⟩ | M is a TM that accepts exactly two strings }

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

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

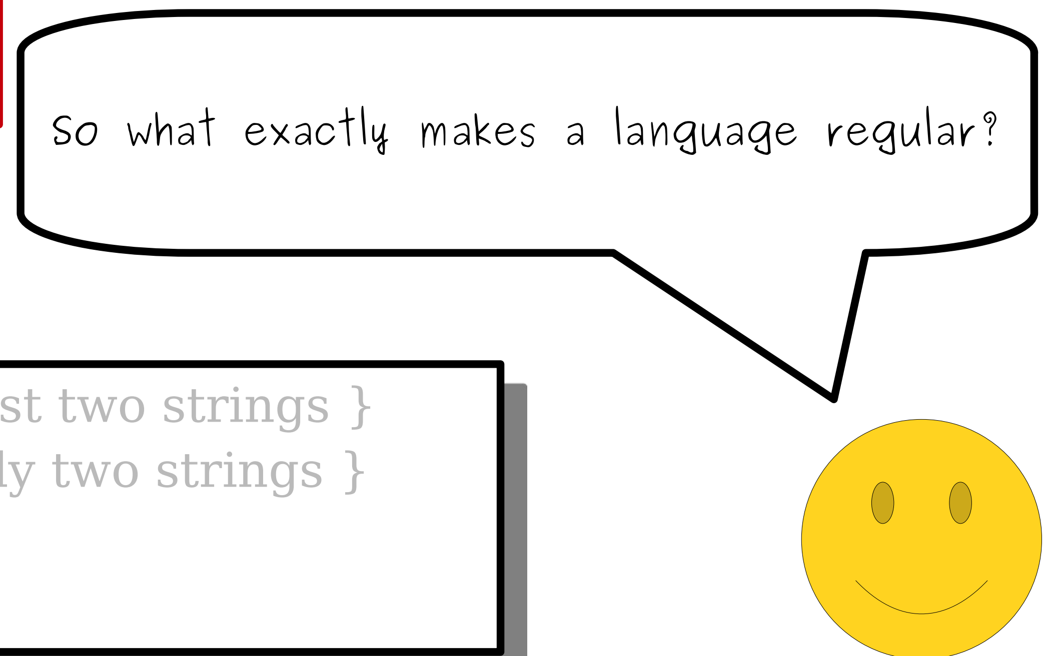
RE: Languages with Verifiers

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 $w \notin L$?



$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

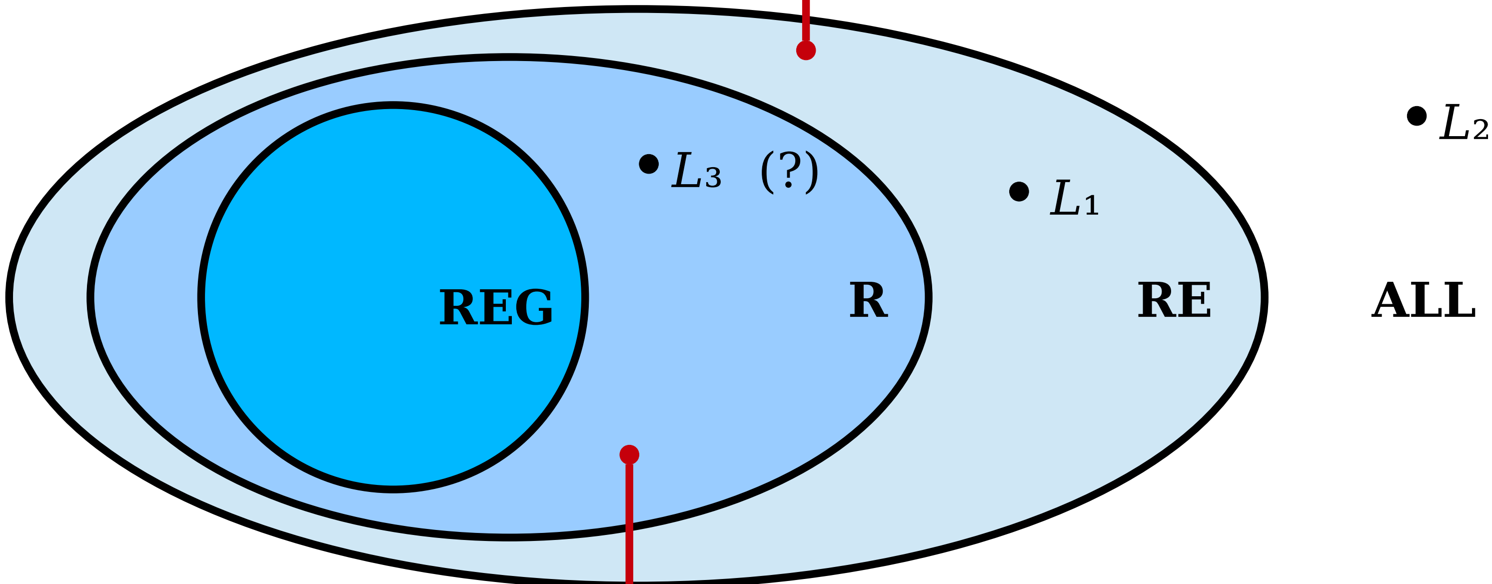
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

$L_4 = \{ \texttt{a}^n\texttt{b}^n \mid n \in \mathbb{N}$ 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 $w \notin L$?

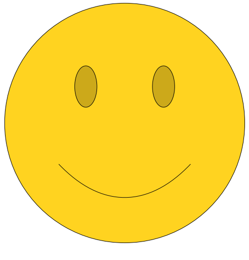
We have a ton of different definitions for regular languages – they're the languages of DFAs, NFAs, and regexes.

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

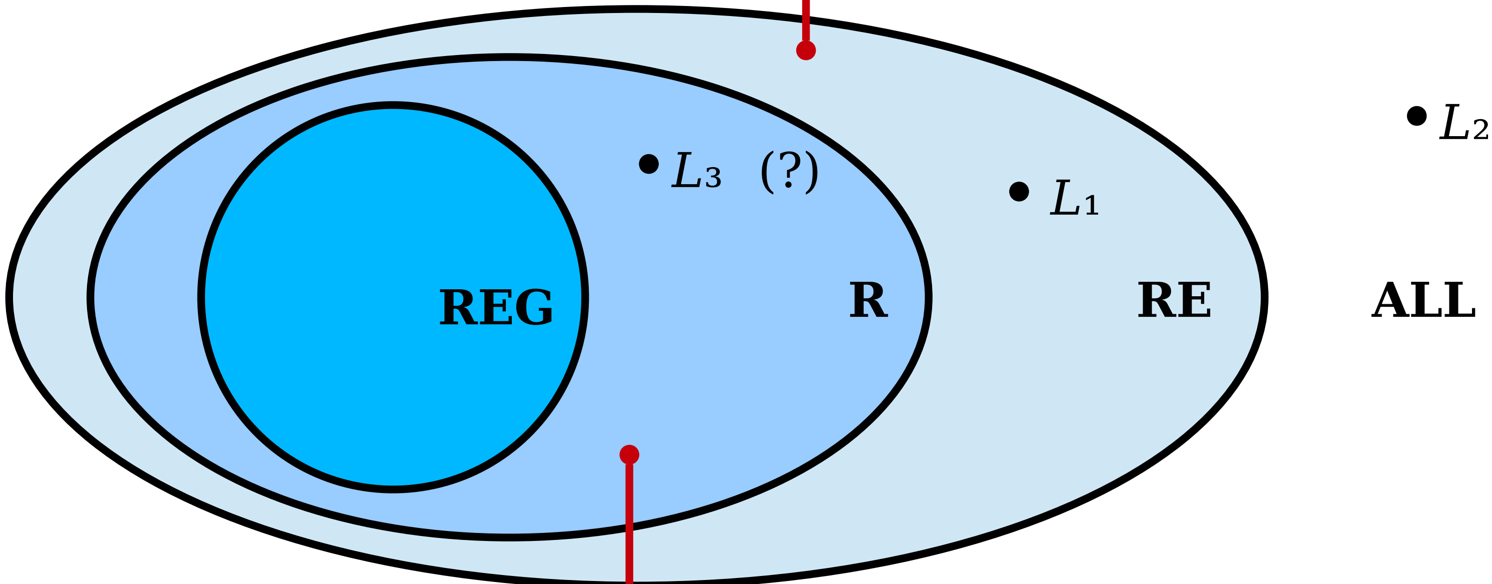
$L_3 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}$ and $n > 1000 \}$

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



RE: Languages with Verifiers

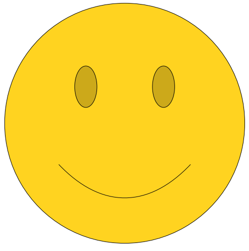
Given any string $w \in L$, could you ***prove*** that $w \in L$?



R: Languages with Deciders

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

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.



$L_1 = \{ \langle M \rangle \mid M \text{ is a TM that accepts at least two strings} \}$

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM that accepts exactly two strings} \}$

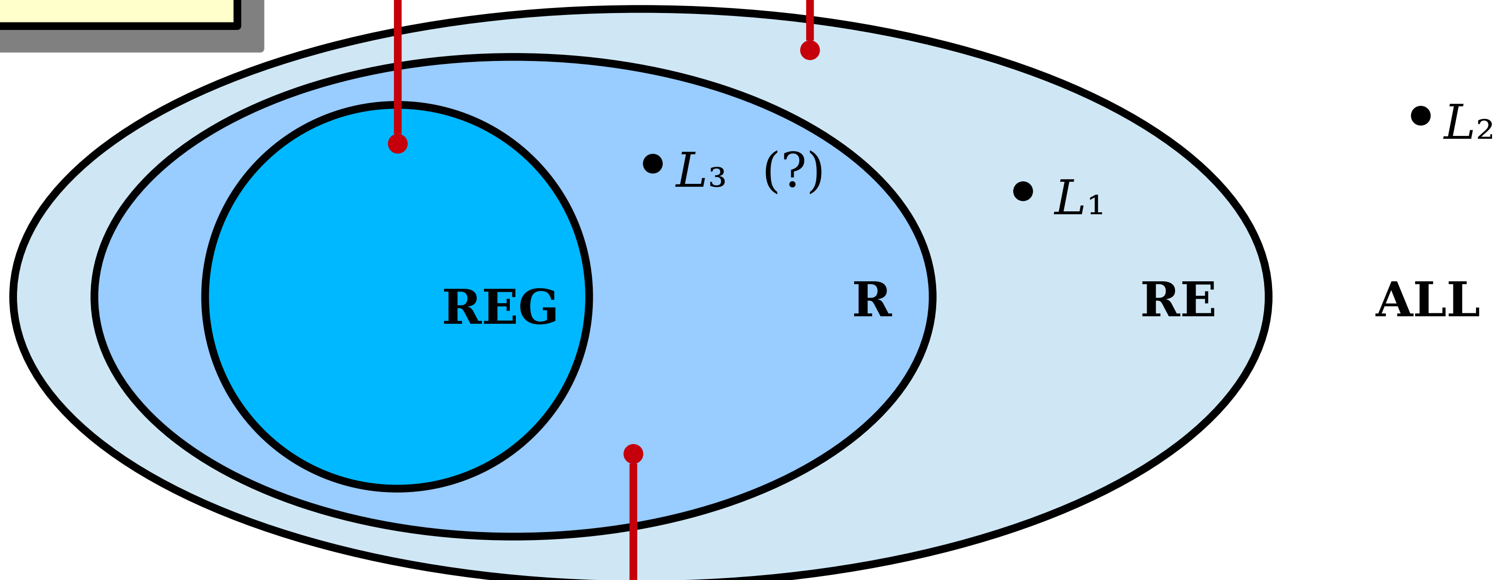
$L_3 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N} \text{ and } n > 1000 \}$

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

REG: Problems Solvable with Finite Memory

RE: Languages with Verifiers

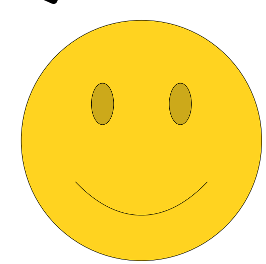
Given any string $w \in L$, could you ***prove*** that $w \in L$?



R: Languages with Deciders

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

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.)

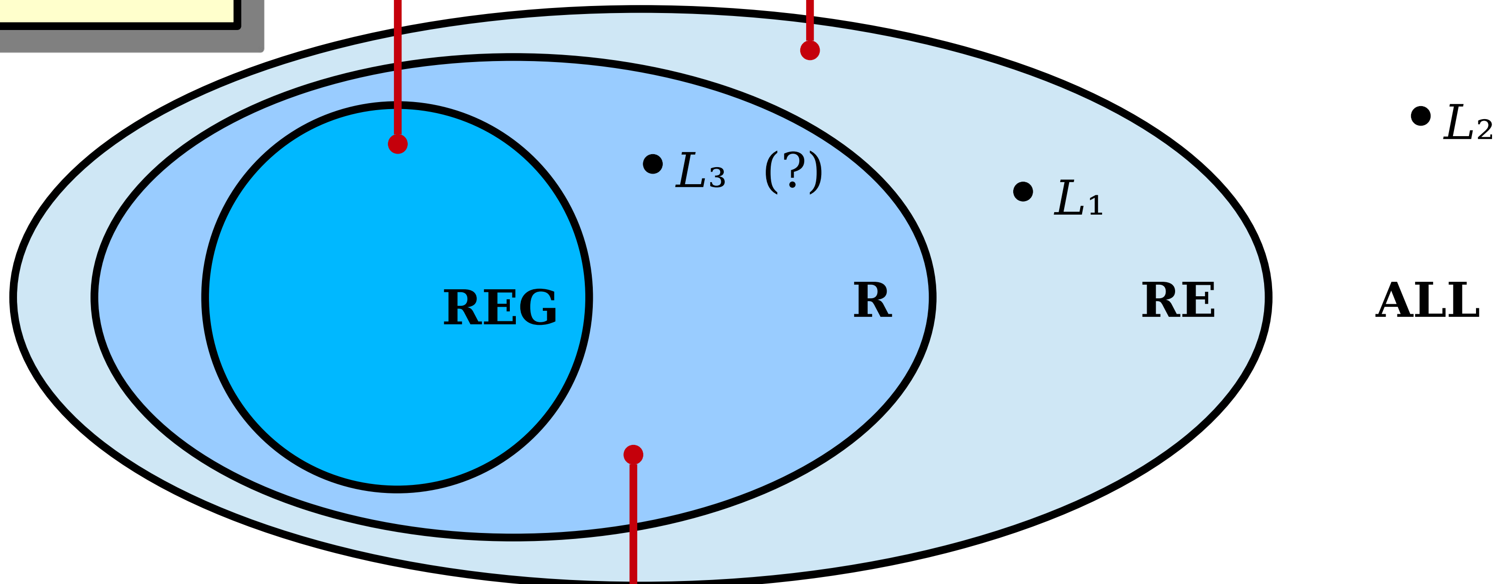


$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$
$L_3 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}$ and $n > 1000 \}$
$L_4 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}$ and $n \leq 1000 \}$

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.

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

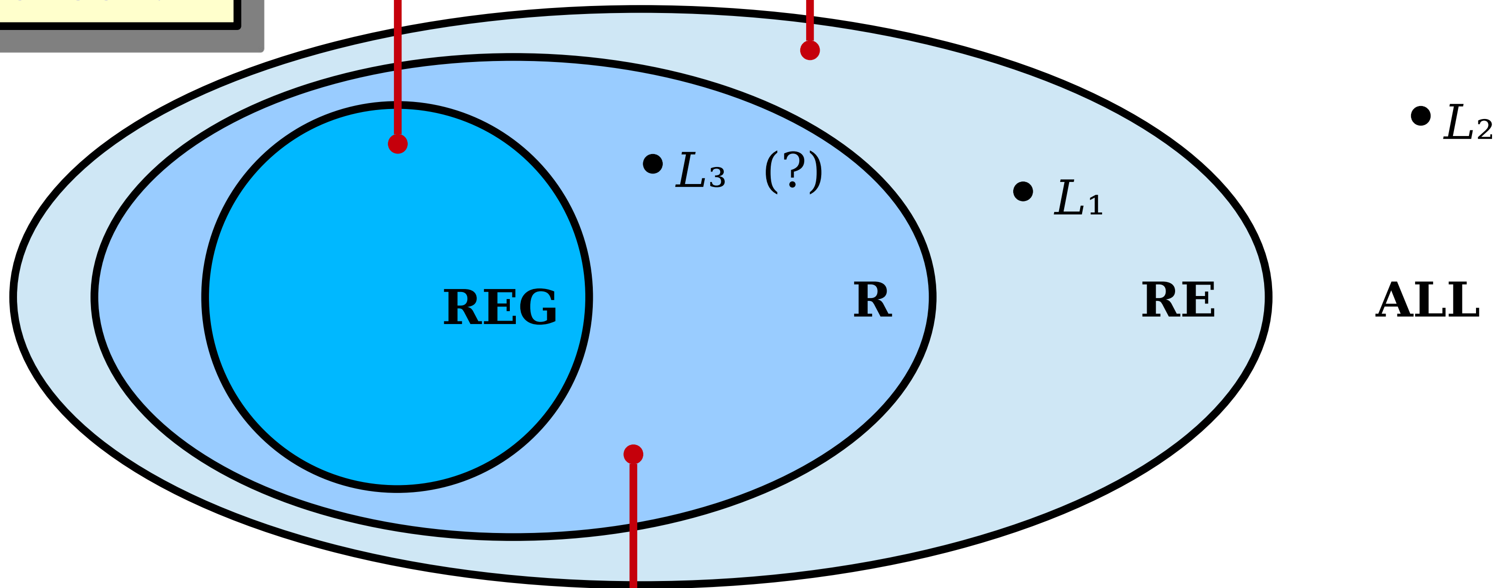
$L_4 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}$ and $n \leq 1000 \}$

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.

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

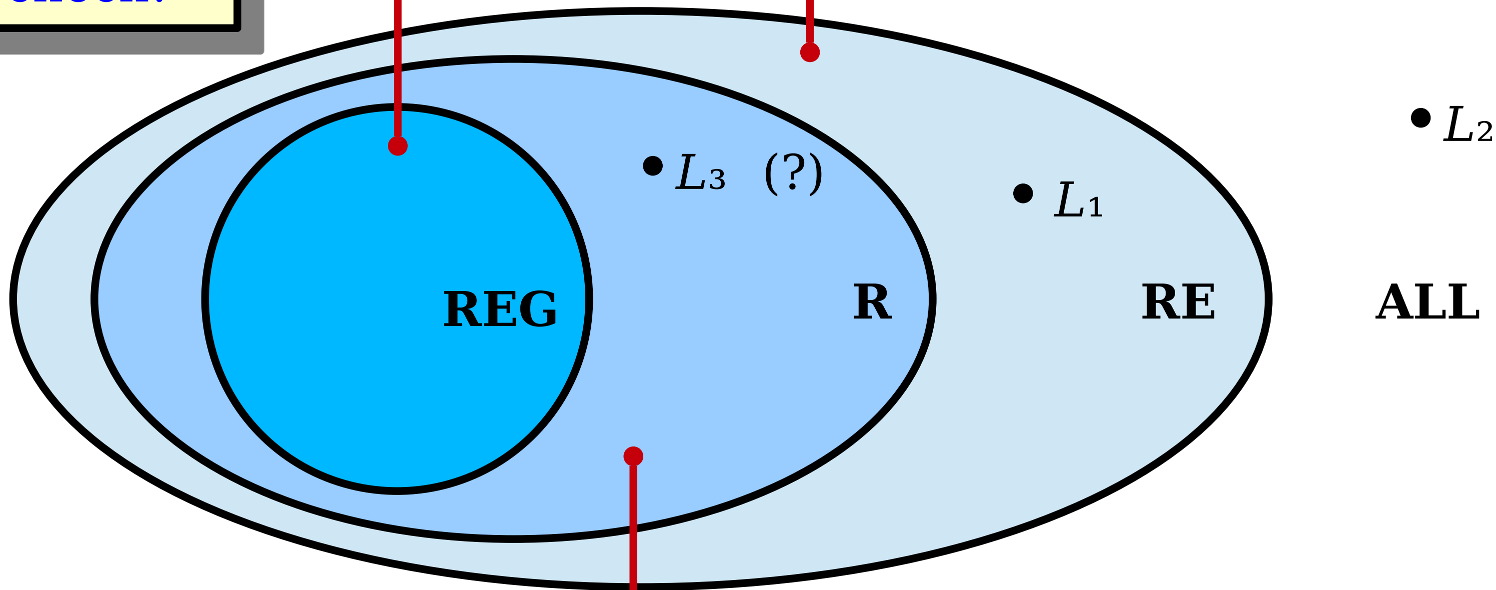
$L_4 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}$ and $n \leq 1000 \}$

REG: Problems Solvable with Finite Memory

Are there are finitely many cases to check?

RE: Languages with Verifiers

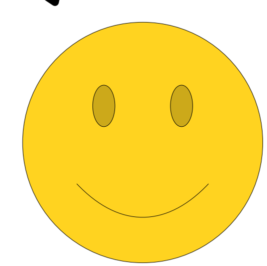
Given any string $w \in L$, could you ***prove*** that $w \in L$?



R: Languages with Deciders

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

So let's think about this here. What information do we need to keep track of?



$L_1 = \{ \langle M \rangle \mid M \text{ is a TM that accepts at least two strings } \}$

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM that accepts exactly two strings } \}$

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

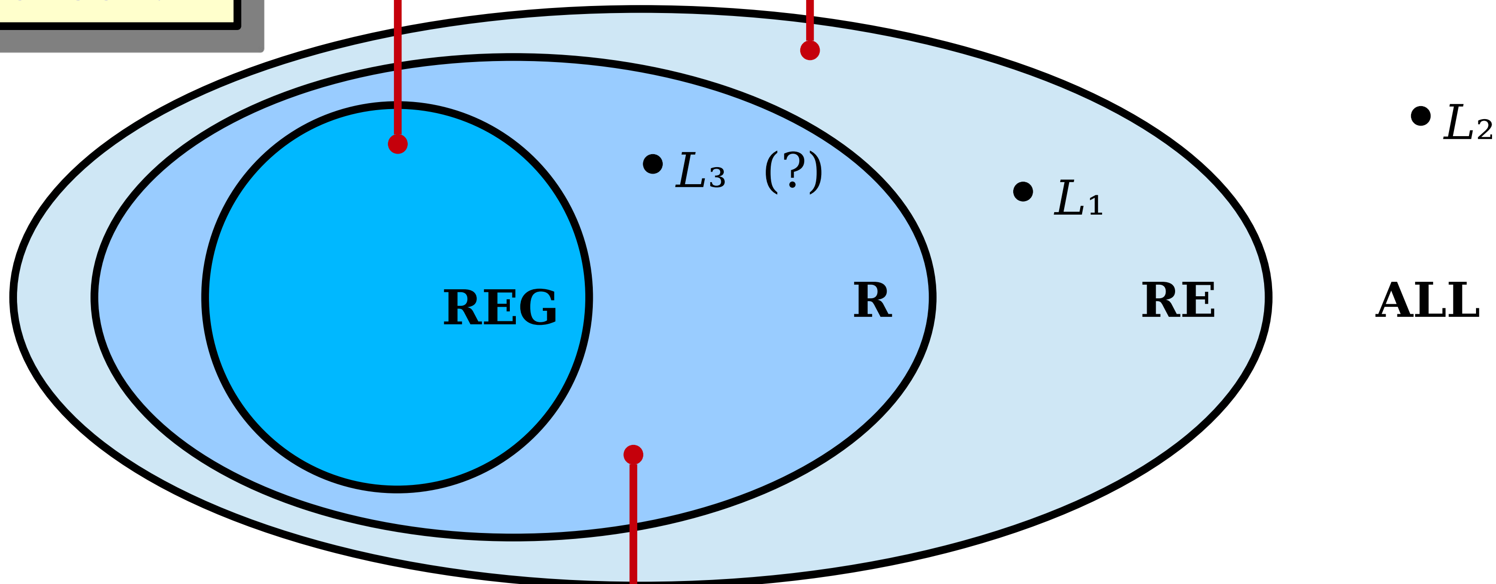
$L_4 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$

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 $w \notin L$?

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.

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

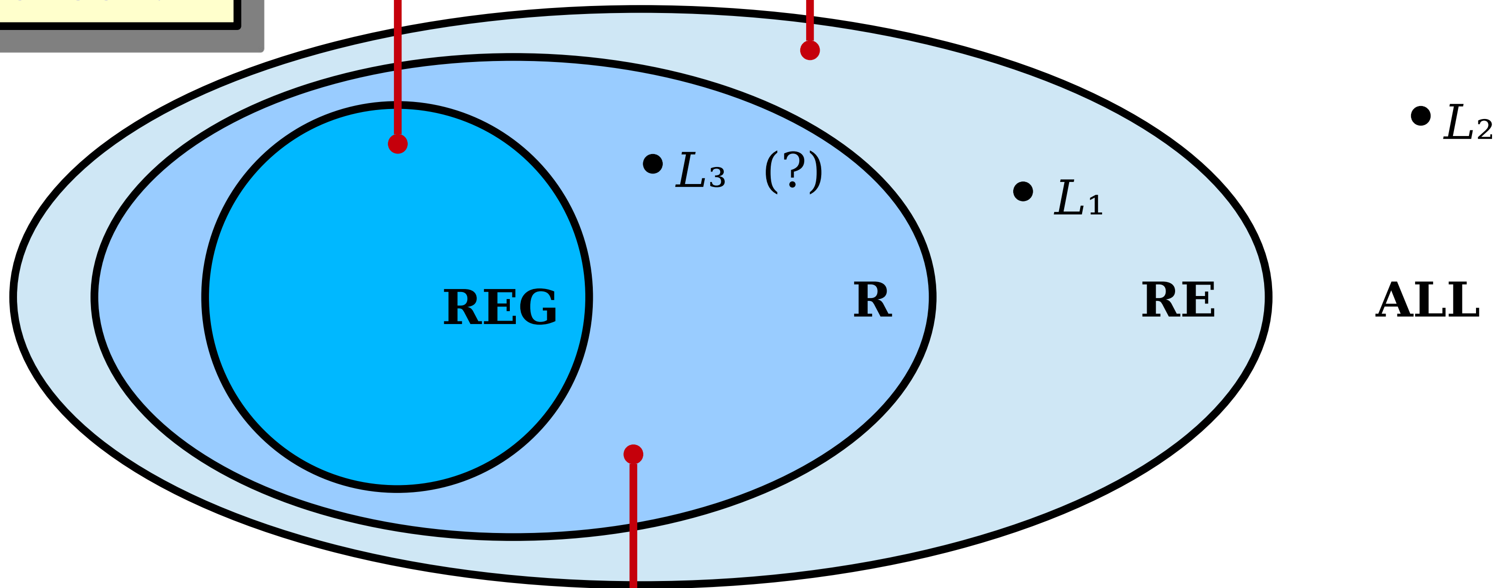
$L_4 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}$ and $n \leq 1000 \}$

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 $w \notin L$?

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!

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

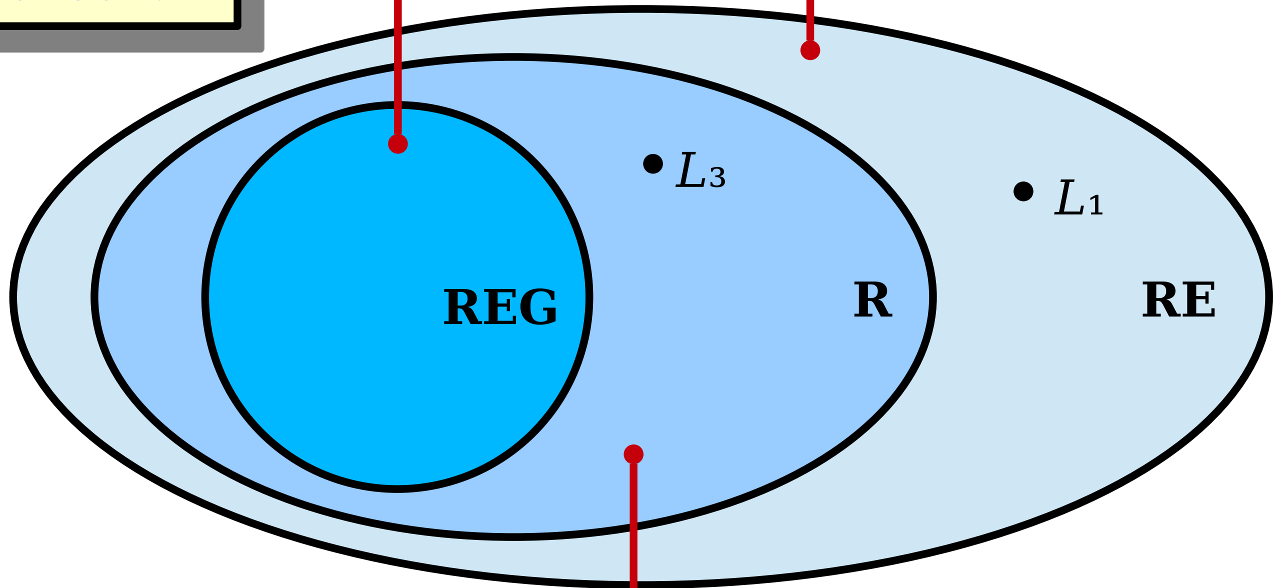
$L_4 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}$ and $n \leq 1000 \}$

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 $w \notin L$?

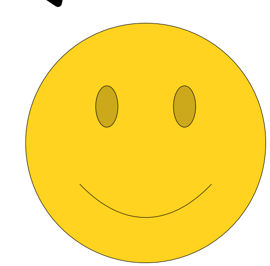
So this gives us the intuition that L_3 is almost certainly going to be nonregular.

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

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

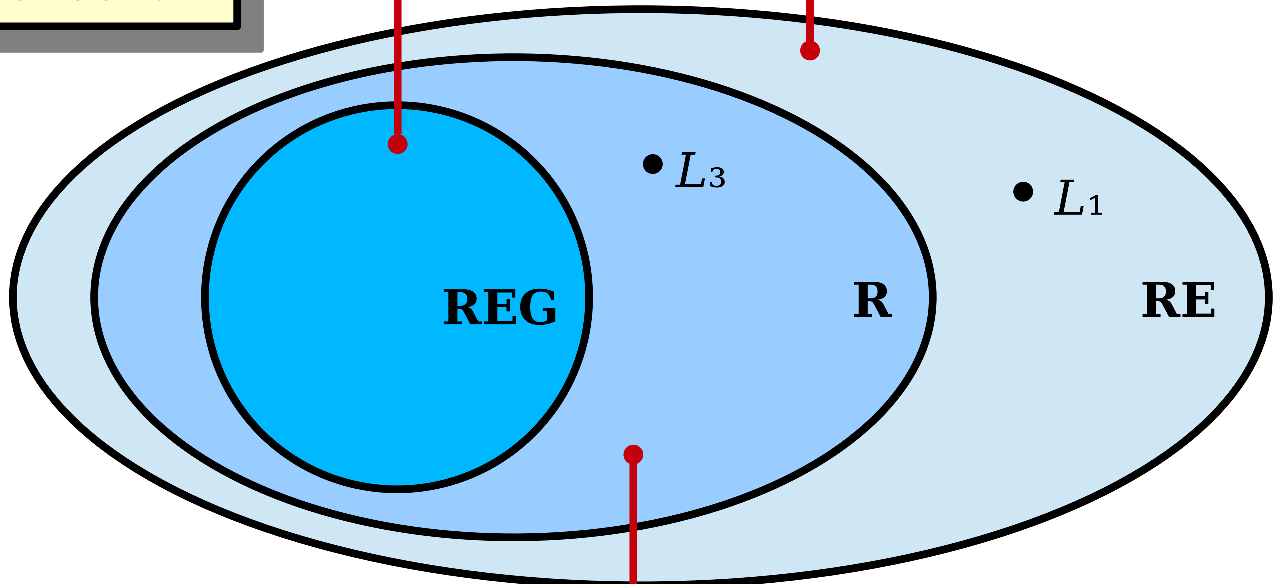


REG: Problems Solvable with Finite Memory

Are there are finitely many cases to check?

RE: Languages with Verifiers

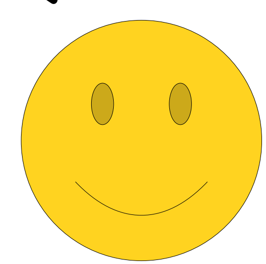
Given any string $w \in L$, could you ***prove*** that $w \in L$?



R: Languages with Deciders

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

You can formally prove this by using the Myhill–Nerode theorem. I highly recommend it – it's good practice!



$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

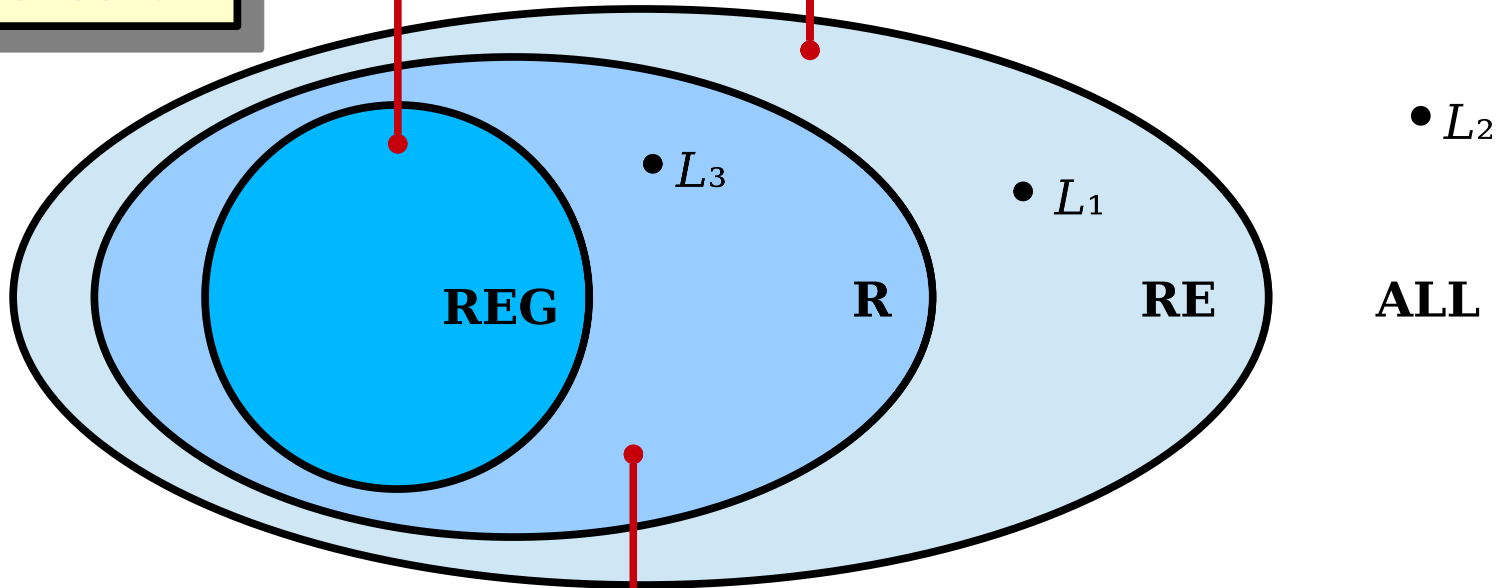
$L_4 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}$ and $n \leq 1000 \}$

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 $w \notin L$?

so – what did we learn here?

L_1 = { ⟨M⟩ | M is a TM that accepts at least two strings }

L_2 = { ⟨M⟩ | M is a TM that accepts exactly two strings }

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

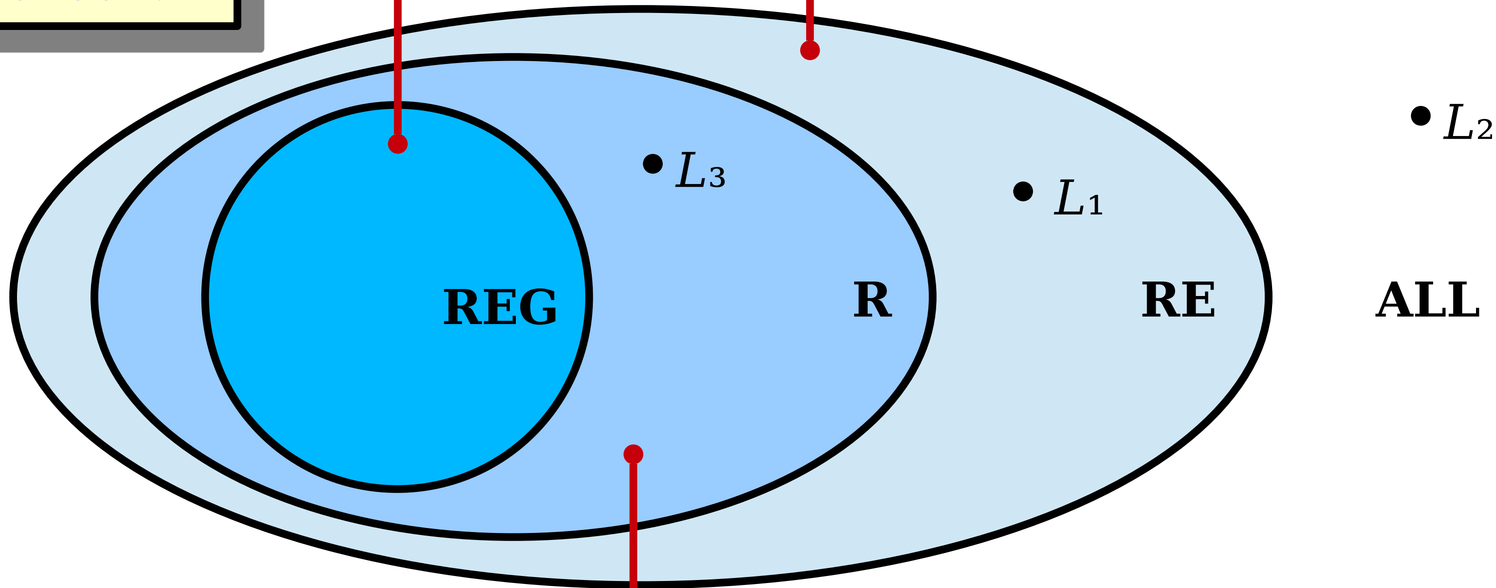
$L_4 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$

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 $w \notin L$?

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.

L_1 = { ⟨*M*⟩ | *M* is a TM that accepts at least two strings }

L_2 = { ⟨*M*⟩ | *M* is a TM that accepts exactly two strings }

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

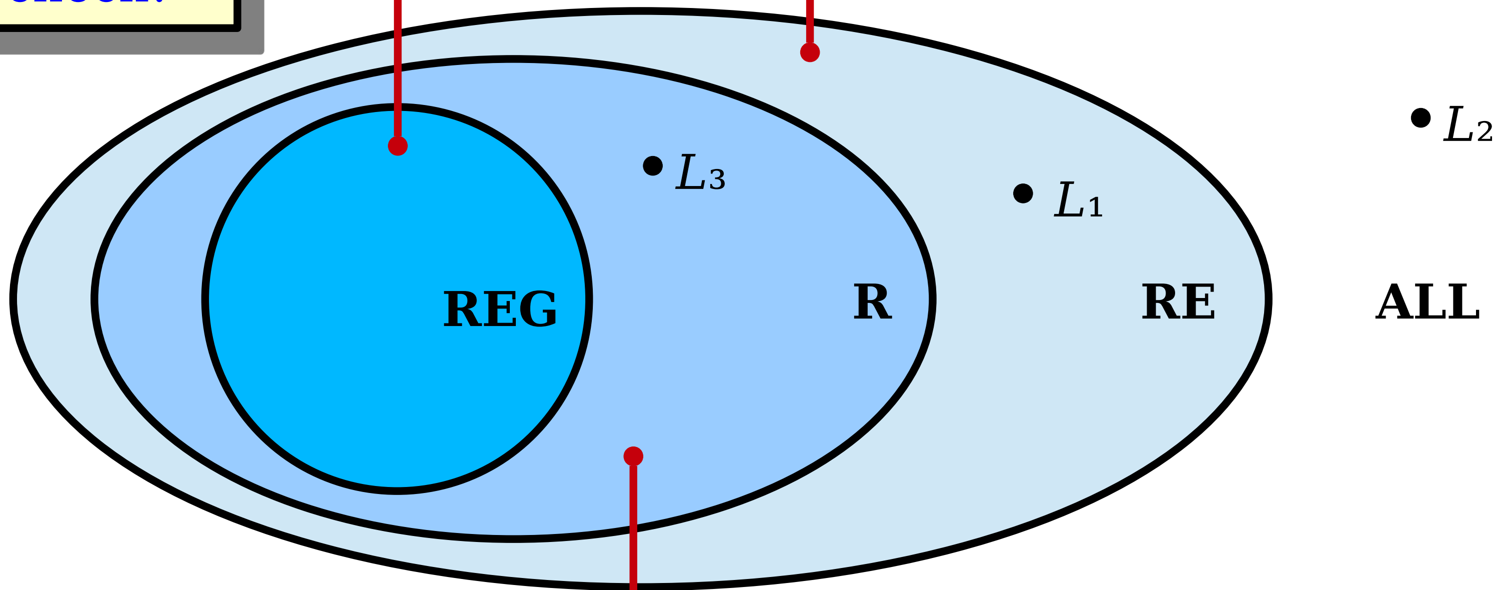
$L_4 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$

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 $w \notin L$?

With all that said and done, let's move on to our last language here.

L_1 = { ⟨M⟩ | M is a TM that accepts at least two strings }

L_2 = { ⟨M⟩ | M is a TM that accepts exactly two strings }

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

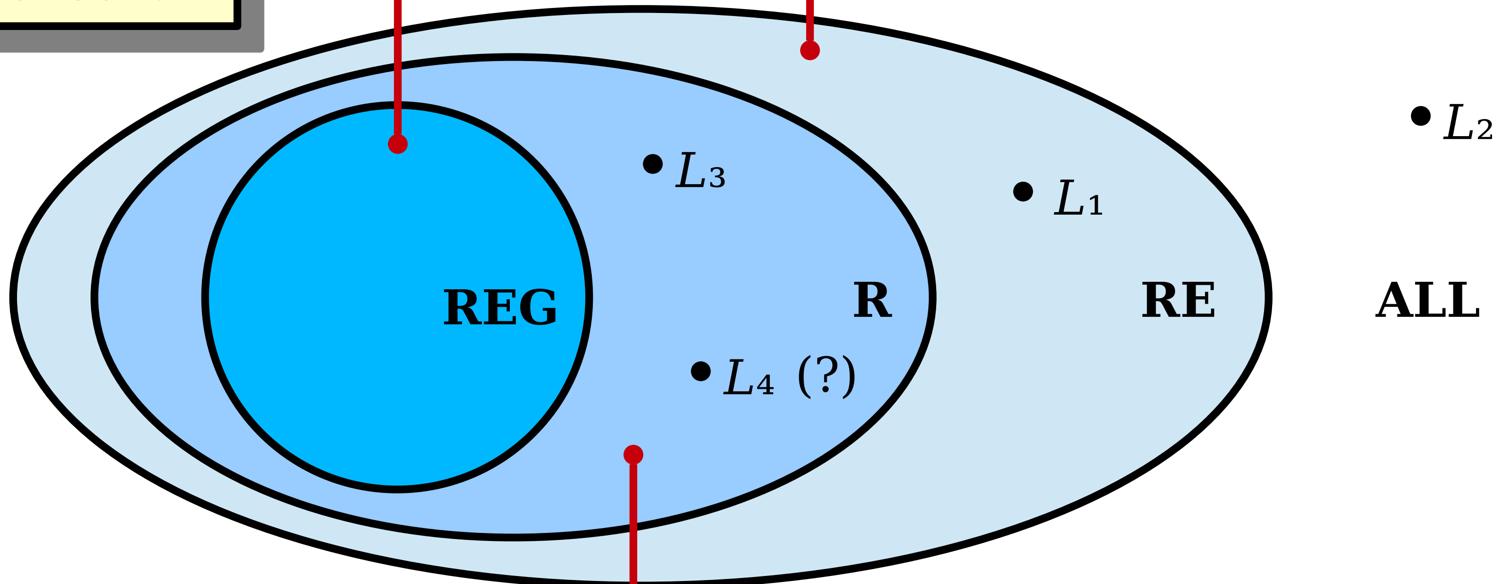
L_4 = { $\mathtt{a}^n\mathtt{b}^n$ | $n \in \mathbb{N}$ and $n \leq 1000$ }

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 $w \notin L$?

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.

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

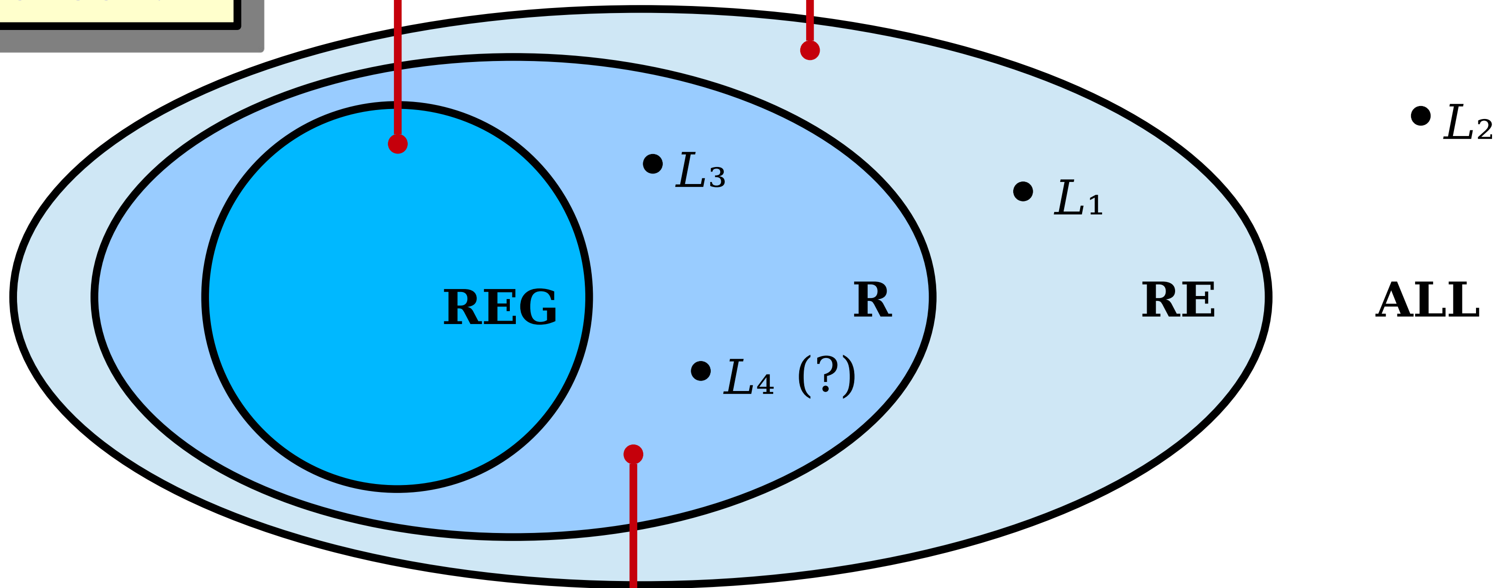
$L_4 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}$ and $n \leq 1000 \}$

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 $w \notin L$?

The question now is whether it's regular or not.

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

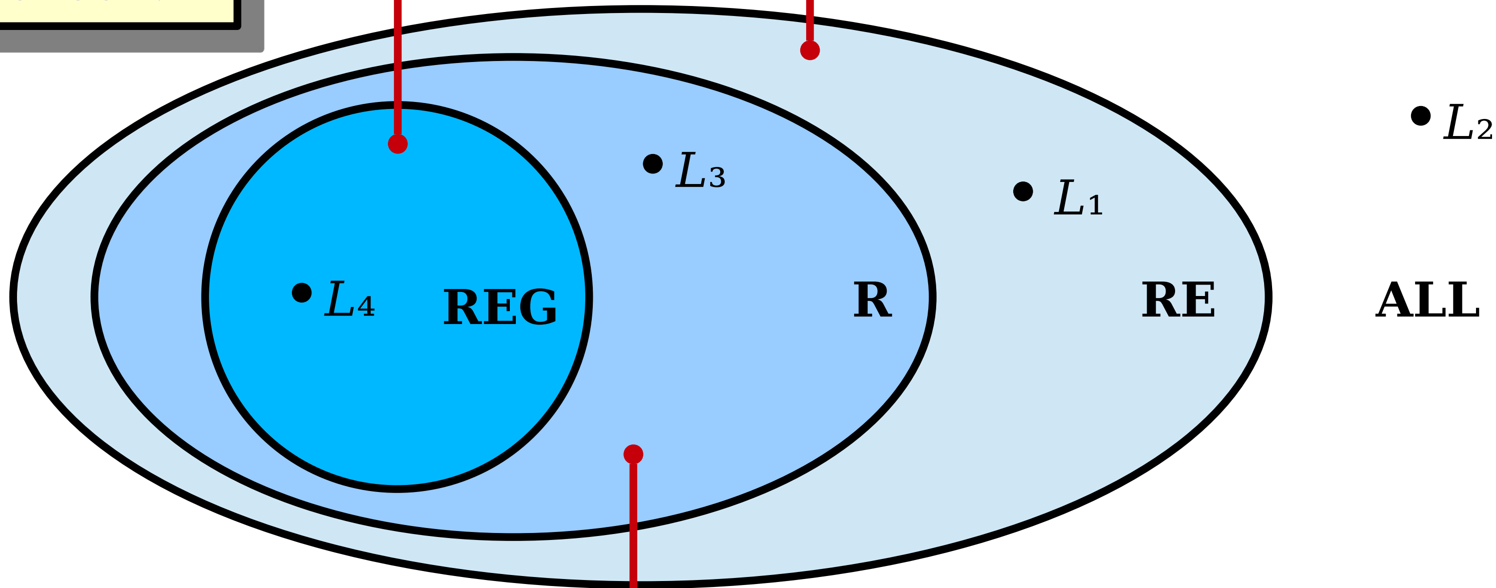
$L_4 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}$ and $n \leq 1000 \}$

REG: Problems Solvable with Finite Memory

Are there are finitely many cases to check?

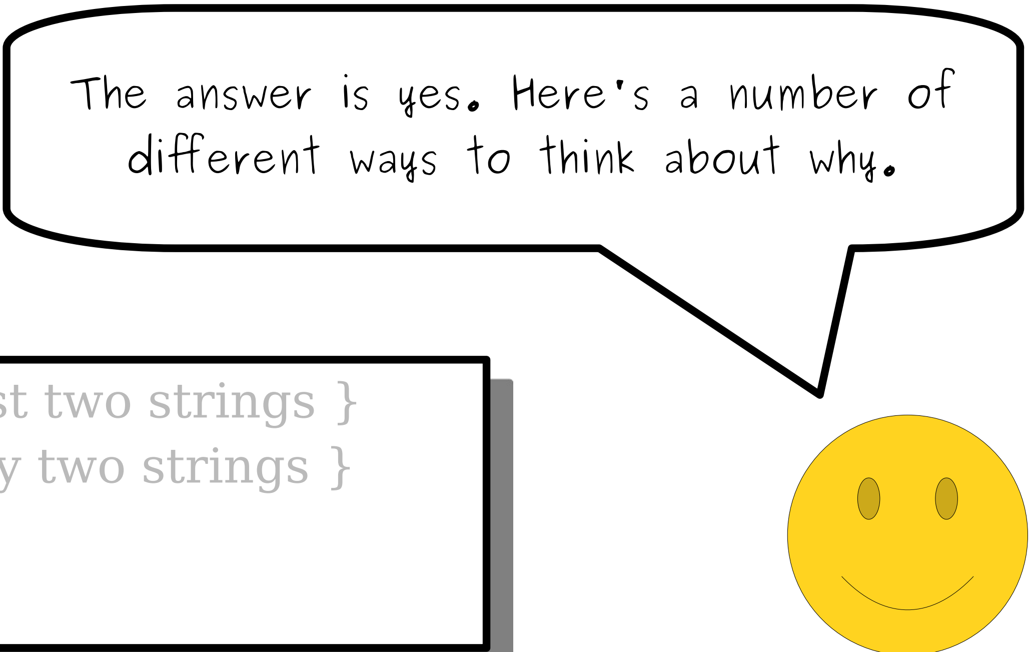
RE: Languages with Verifiers

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 $w \notin L$?

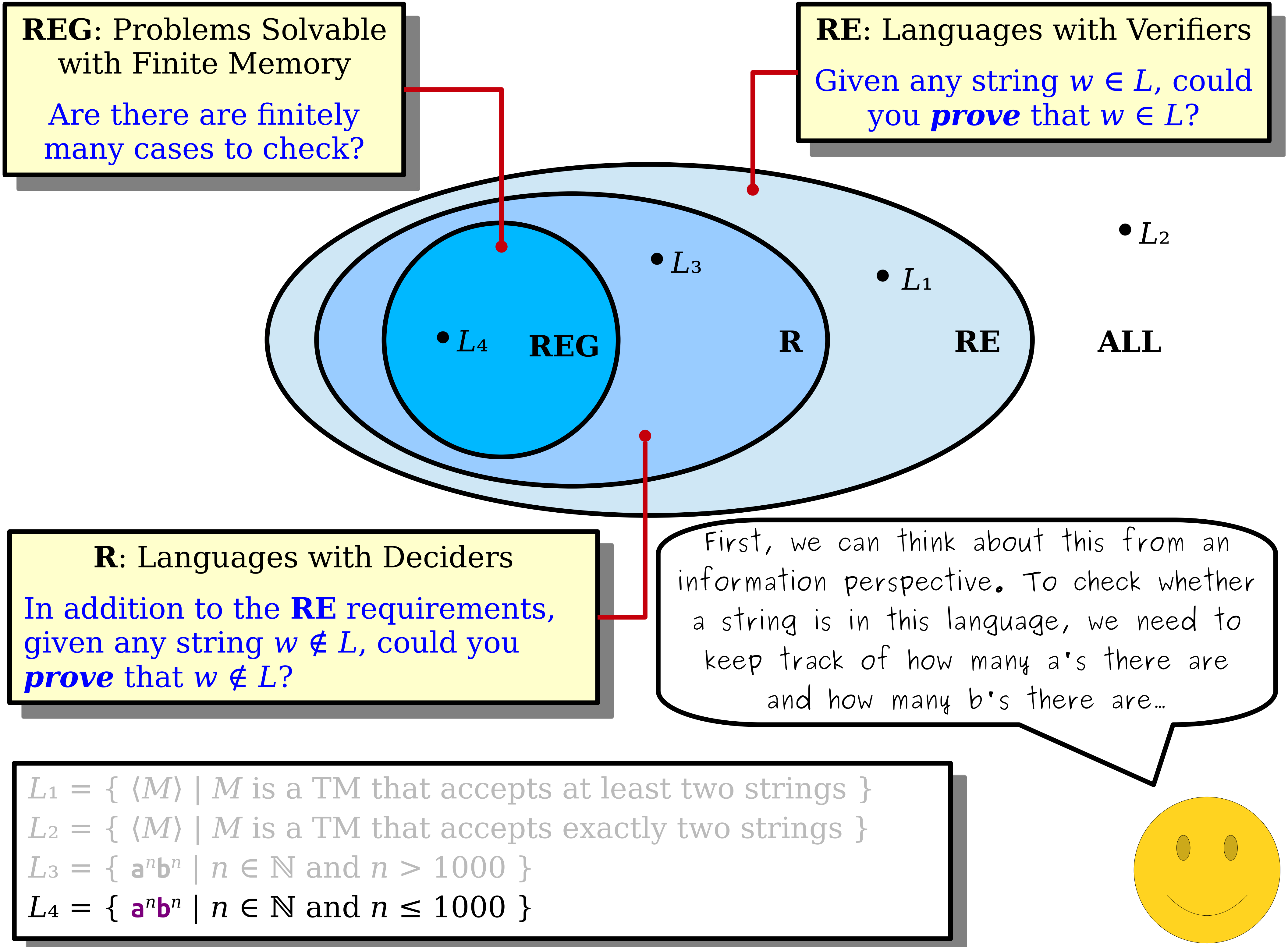


$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

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



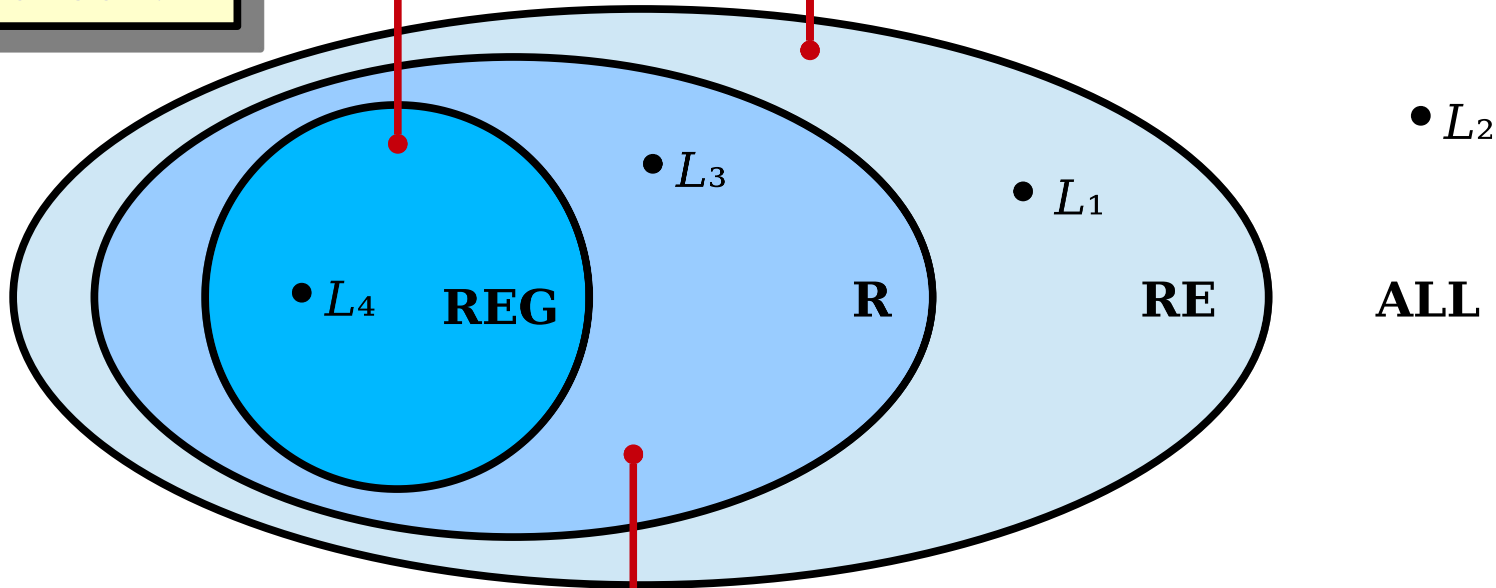
REG: Problems Solvable with Finite Memory
Are there are finitely many cases to check?
RE: Languages with Verifiers
Given any string $w \in L$, could you ***prove*** that $w \in L$?
L_2
L_3
L_1
L_4
REG
R
RE
ALL
R: Languages with Deciders
In addition to the **RE** requirements, given any string $w \notin L$, could you ***prove*** that $w \notin L$?
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 = { ⟨M⟩ | M is a TM that accepts at least two strings }
L_2 = { ⟨M⟩ | M is a TM that accepts exactly two strings }
L_3 = { $\mathtt{a}^n\mathtt{b}^n$ | $n \in \mathbb{N}$ and $n > 1000$ }
L_4 = { $\mathtt{a}^n\mathtt{b}^n$ | $n \in \mathbb{N}$ 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 $w \notin L$?

…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.

$L_1 = \{ \langle M \rangle \mid M \text{ is a TM that accepts at least two strings} \}$

$L_2 = \{ \langle M \rangle \mid M \text{ is a TM that accepts exactly two strings} \}$

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

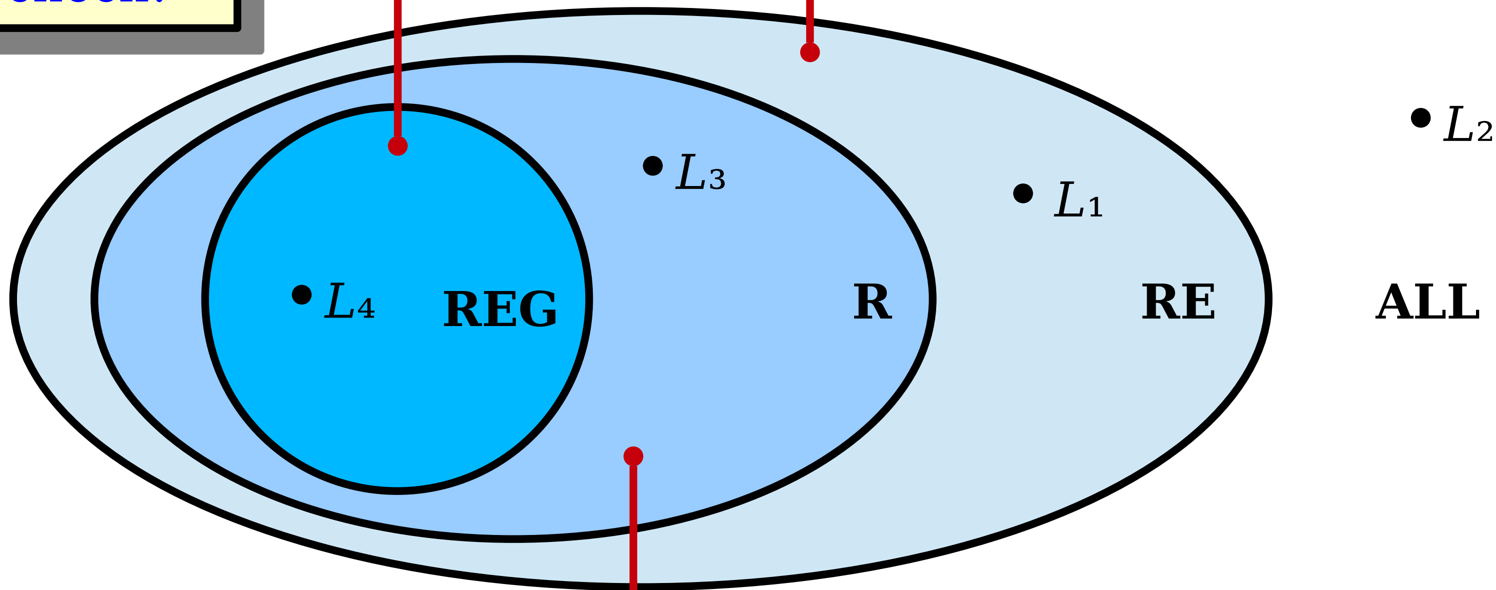
$L_4 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$

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 $w \notin L$?

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.

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

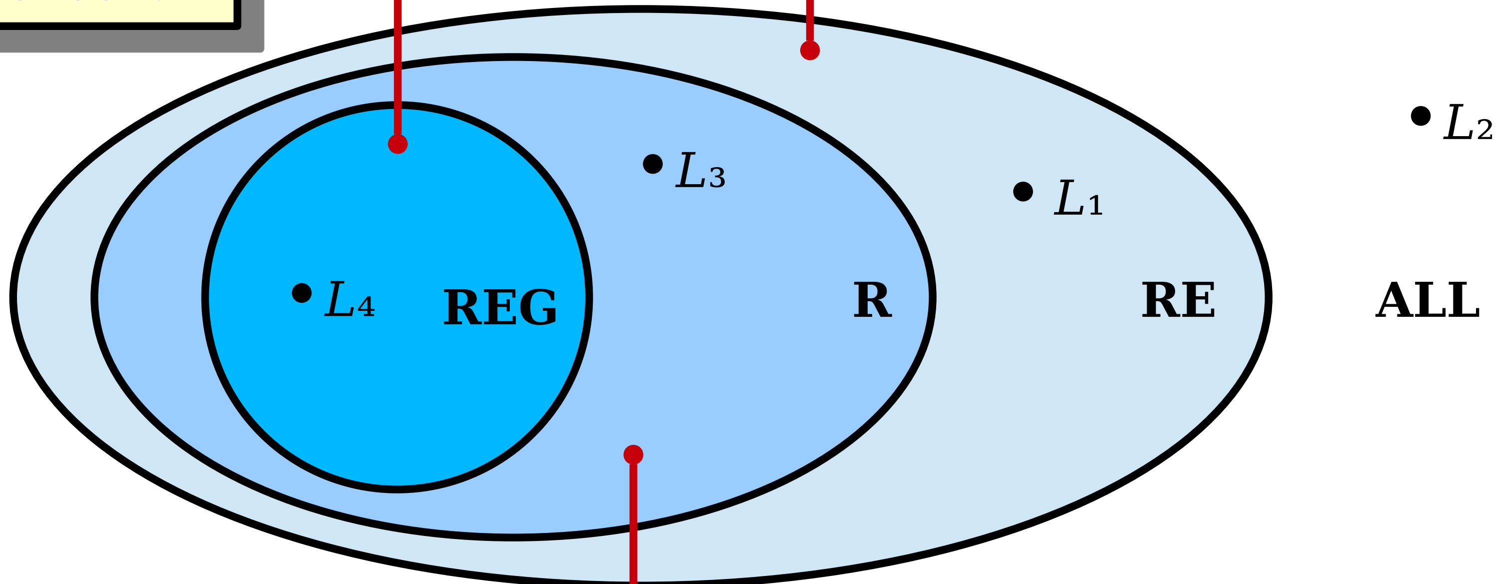
$L_4 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}$ and $n \leq 1000 \}$

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 $w \notin L$?

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 willl be regular.

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

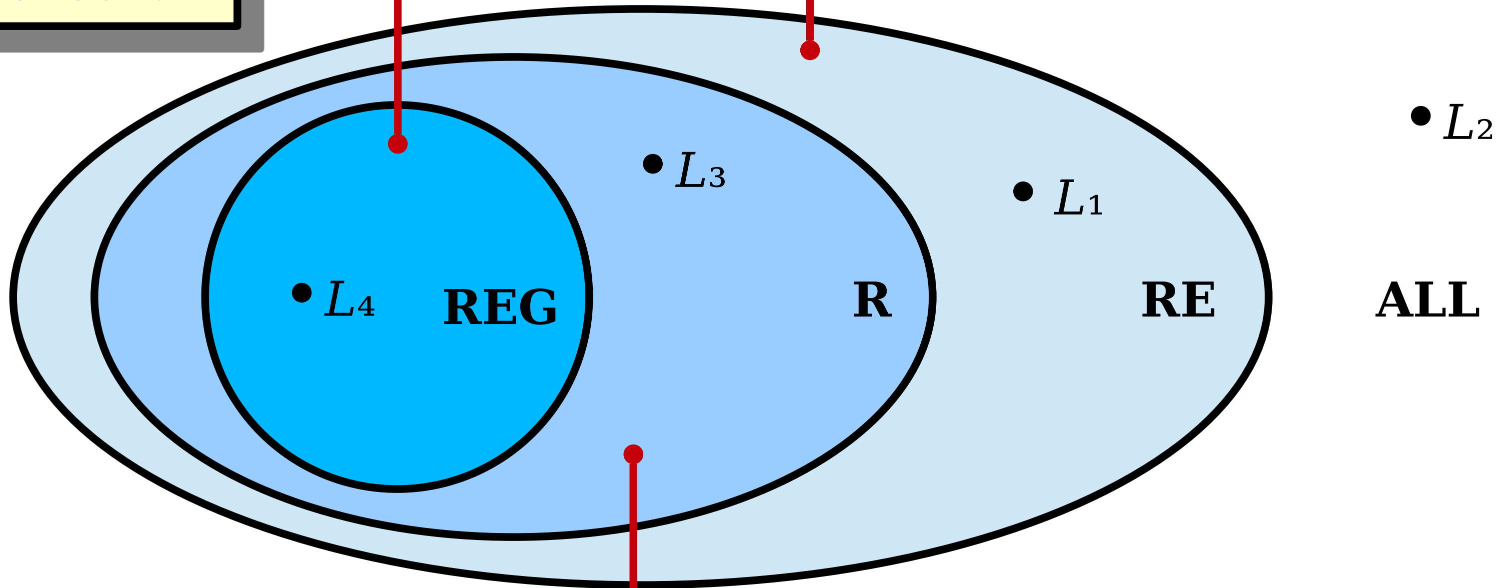
$L_4 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}$ and $n \leq 1000 \}$

REG: Problems Solvable with Finite Memory

Are there are finitely many cases to check?

RE: Languages with Verifiers

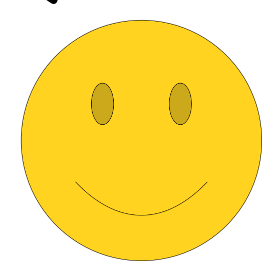
Given any string $w \in L$, could you ***prove*** that $w \in L$?



R: Languages with Deciders

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

Here's another approach we can take.



L_1 = { ⟨M⟩ | M is a TM that accepts at least two strings }

L_2 = { ⟨M⟩ | M is a TM that accepts exactly two strings }

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

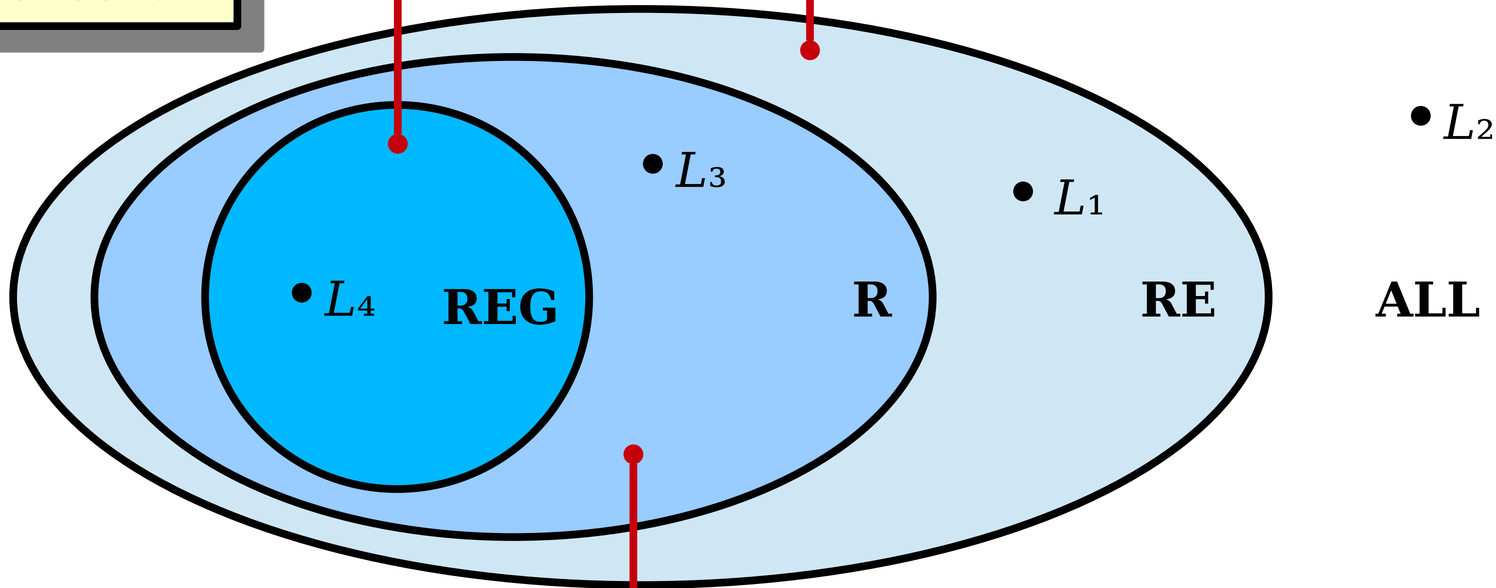
L_4 = { $\mathtt{a}^n\mathtt{b}^n$ | $n \in \mathbb{N}$ and $n \leq 1000$ }

REG: Problems Solvable with Finite Memory

Are there are finitely many cases to check?

RE: Languages with Verifiers

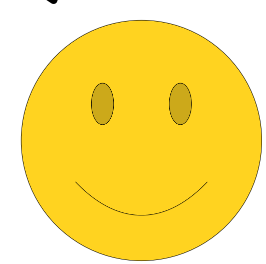
Given any string $w \in L$, could you ***prove*** that $w \in L$?



R: Languages with Deciders

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

How many strings are in this language?



$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

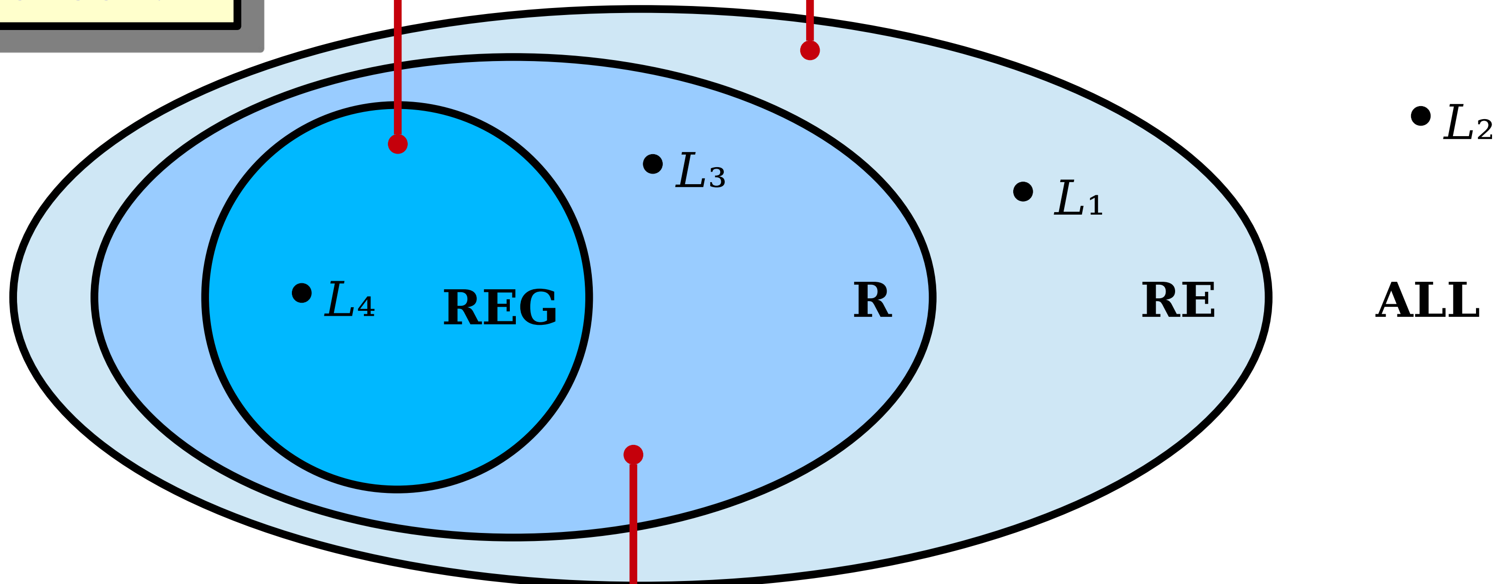
$L_4 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}$ and $n \leq 1000 \}$

REG: Problems Solvable with Finite Memory

Are there are finitely many cases to check?

RE: Languages with Verifiers

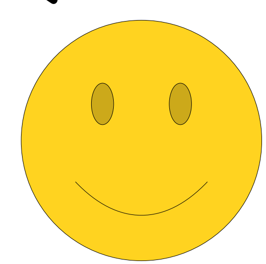
Given any string $w \in L$, could you ***prove*** that $w \in L$?



R: Languages with Deciders

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

There's only 1,001 of them, corresponding to all the different choices of n we can make.



L_1 = { ⟨M⟩ | M is a TM that accepts at least two strings }

L_2 = { ⟨M⟩ | M is a TM that accepts exactly two strings }

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

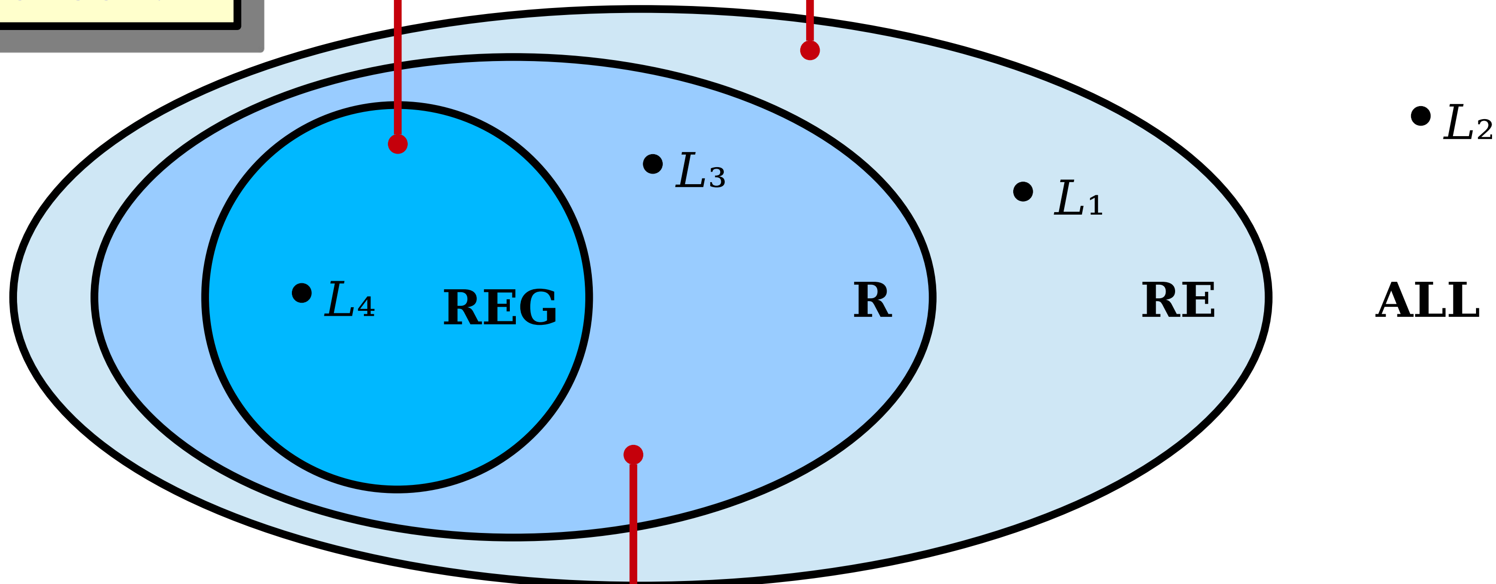
$L_4 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N} \text{ and } n \leq 1000 \}$

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 $w \notin L$?

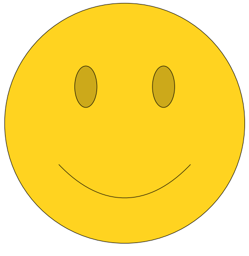
As you proved on Problem Set 7, all finite languages are regular. That means that this language *has* to be regular.

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

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

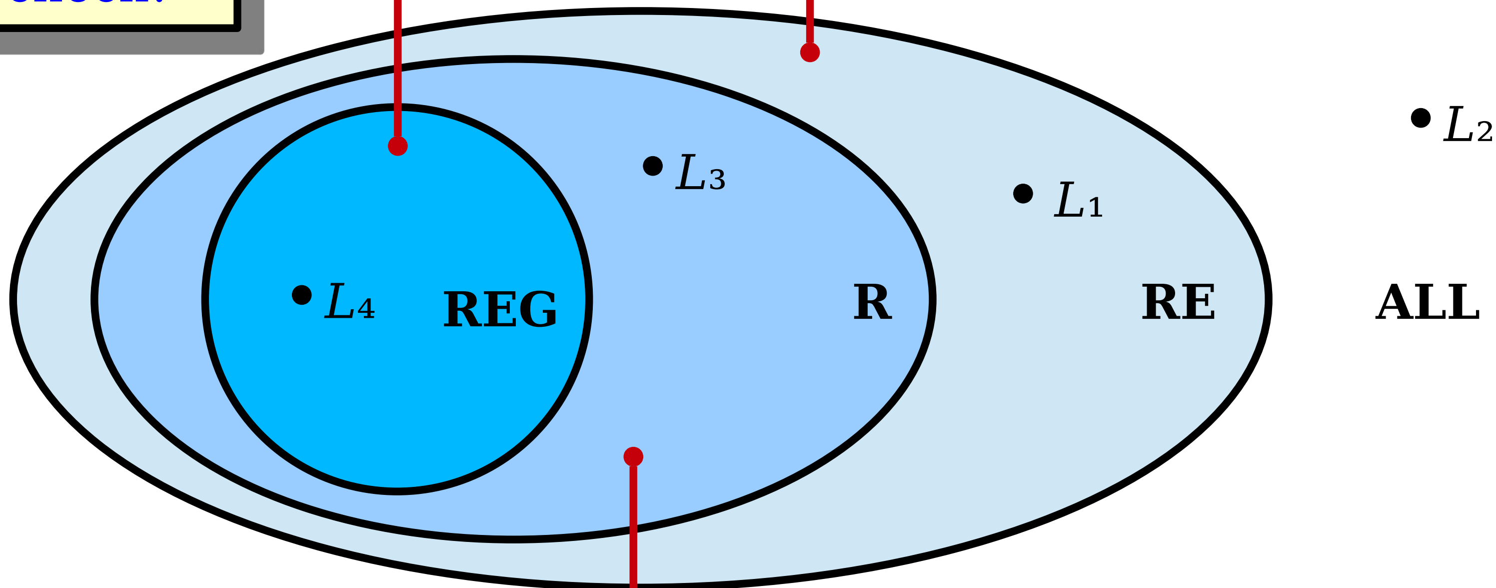


REG: Problems Solvable with Finite Memory

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

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

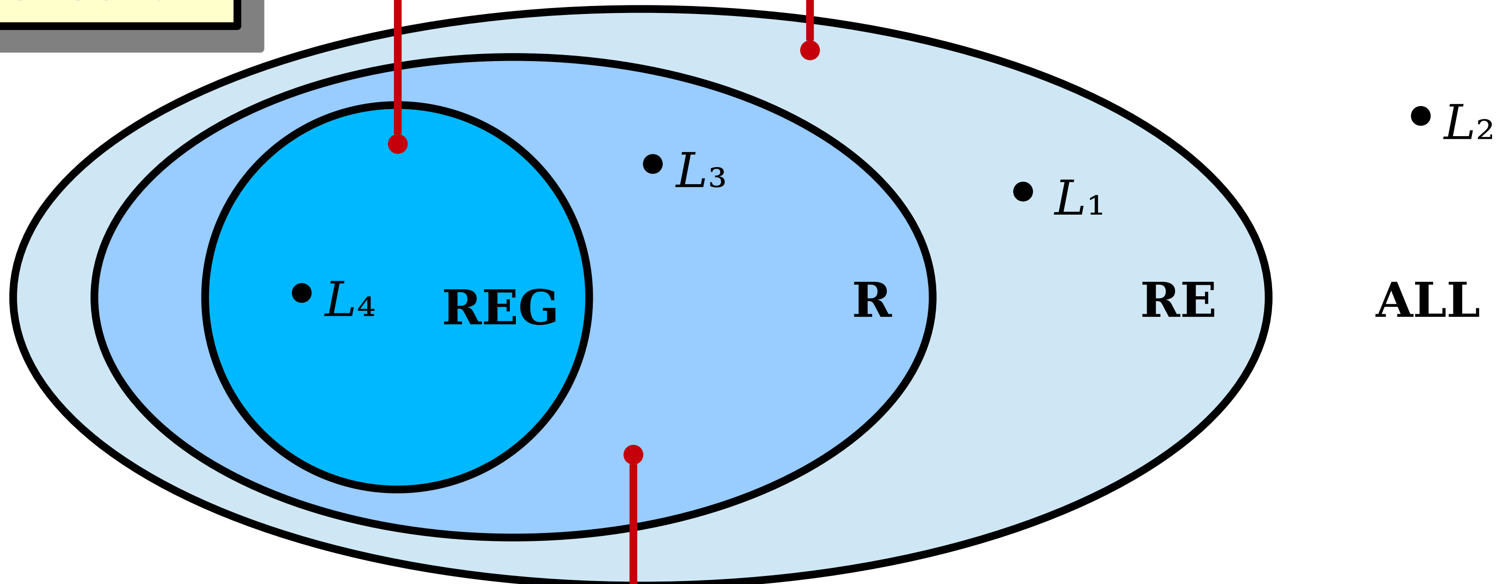
$L_4 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}$ and $n \leq 1000 \}$

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 $w \in L$?

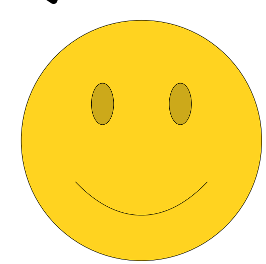


R: Languages with Deciders

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

You could imagine building a (huge) regex for this language:

$\varepsilon \cup ab \cup aabb \cup aaabbb \cup \dots \cup a^{1000}b^{1000}$



$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

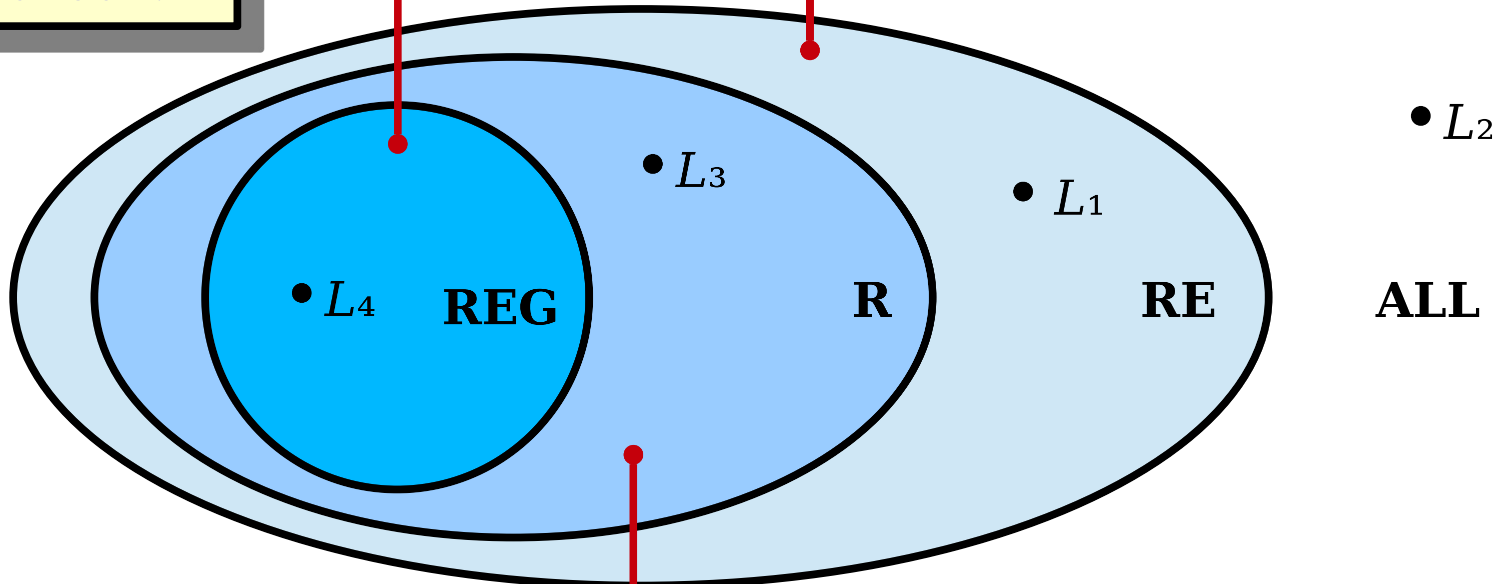
$L_4 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}$ and $n \leq 1000 \}$

REG: Problems Solvable with Finite Memory

Are there are finitely many cases to check?

RE: Languages with Verifiers

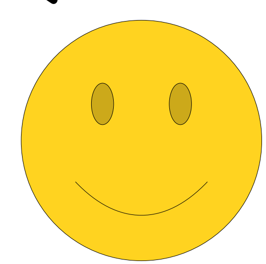
Given any string $w \in L$, could you ***prove*** that $w \in L$?



R: Languages with Deciders

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

So that means that it's going to be regular.



$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

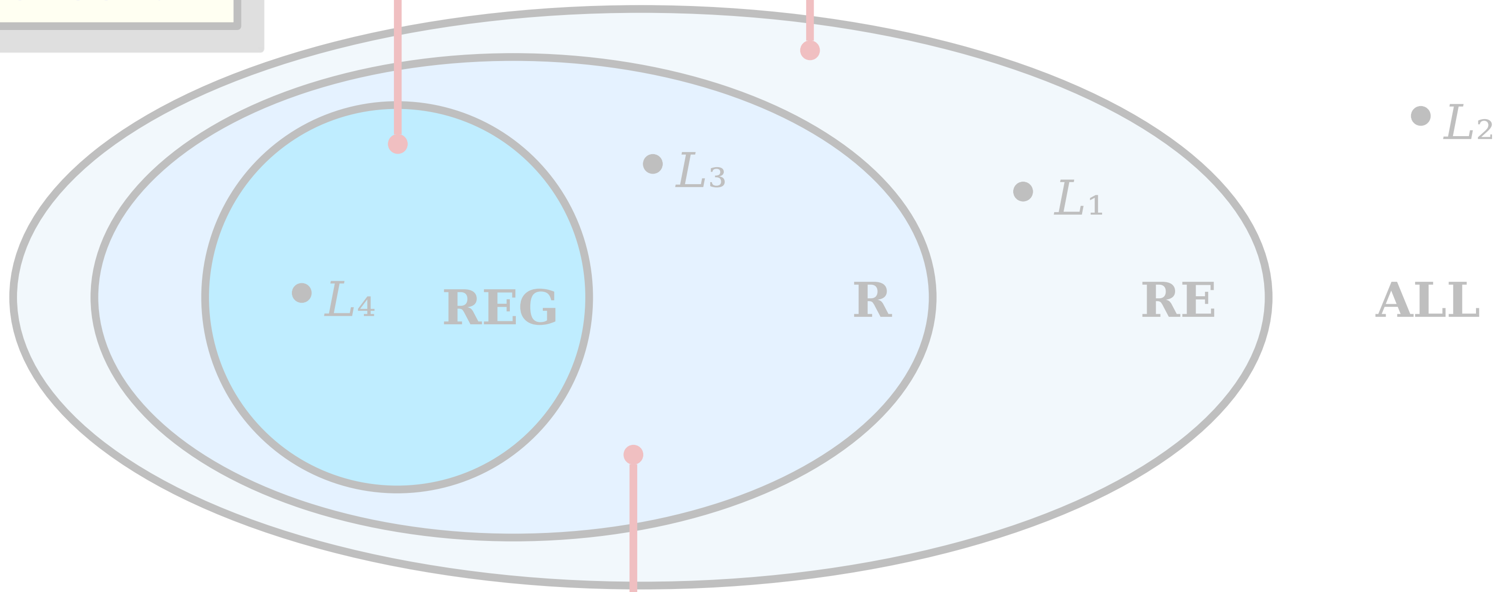
$L_4 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}$ and $n \leq 1000 \}$

REG: Problems Solvable with Finite Memory

Are there are finitely many cases to check?

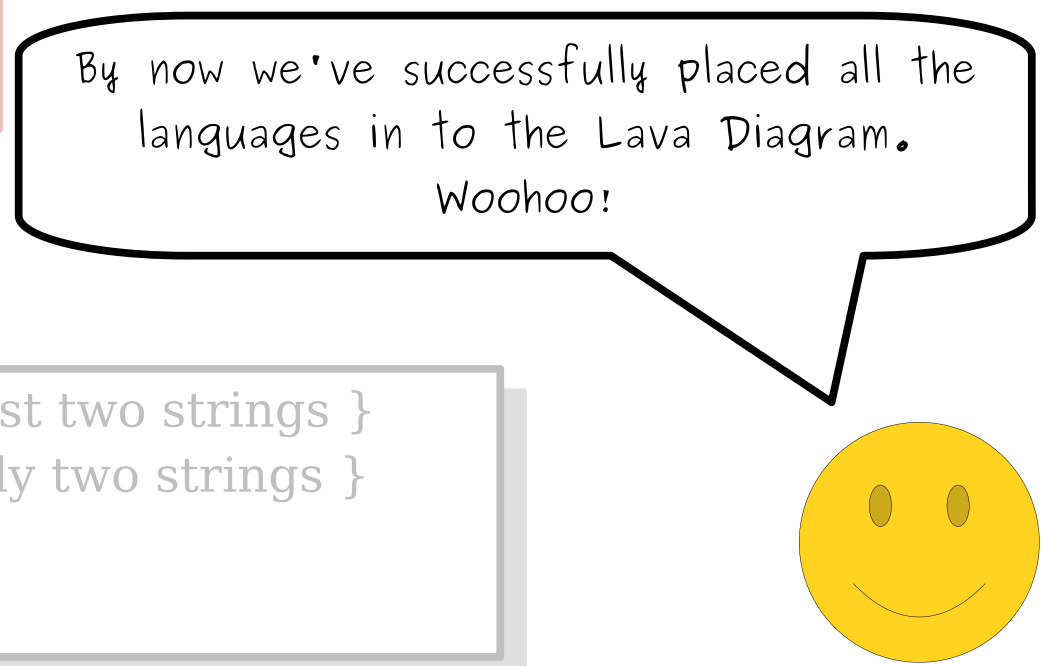
RE: Languages with Verifiers

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 $w \notin L$?



$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

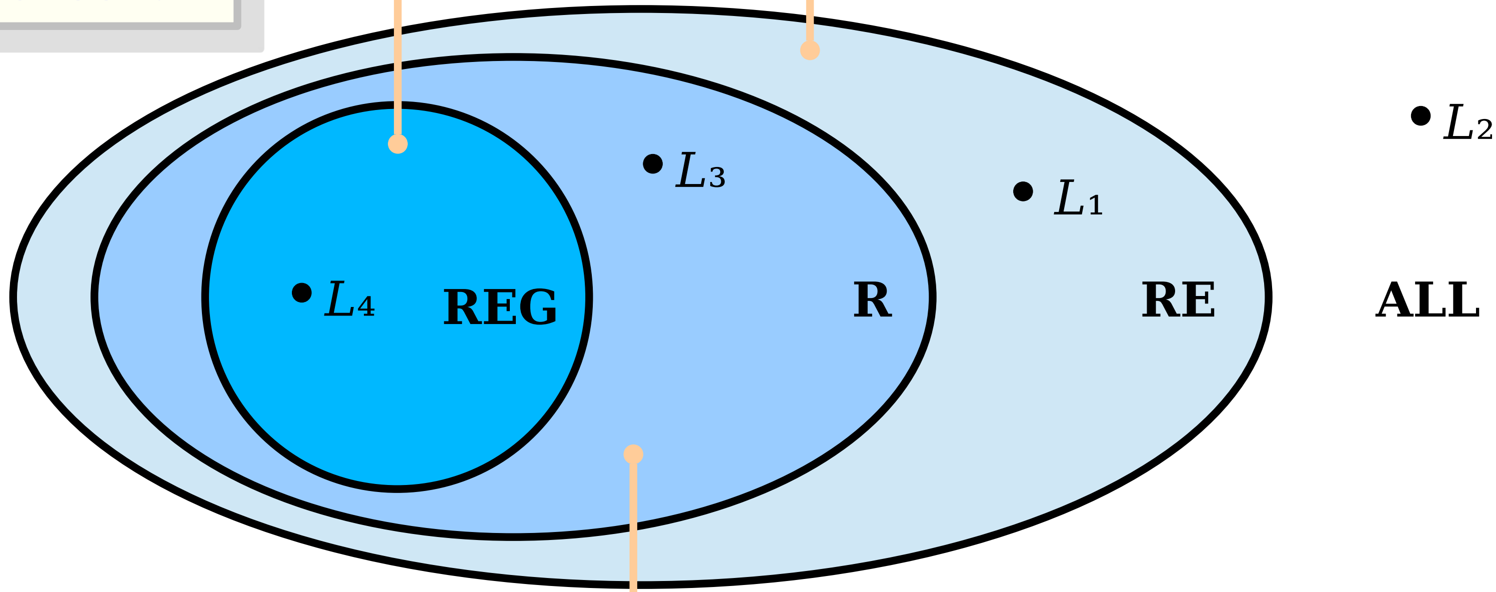
$L_4 = \{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N}$ and $n \leq 1000 \}$

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.

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

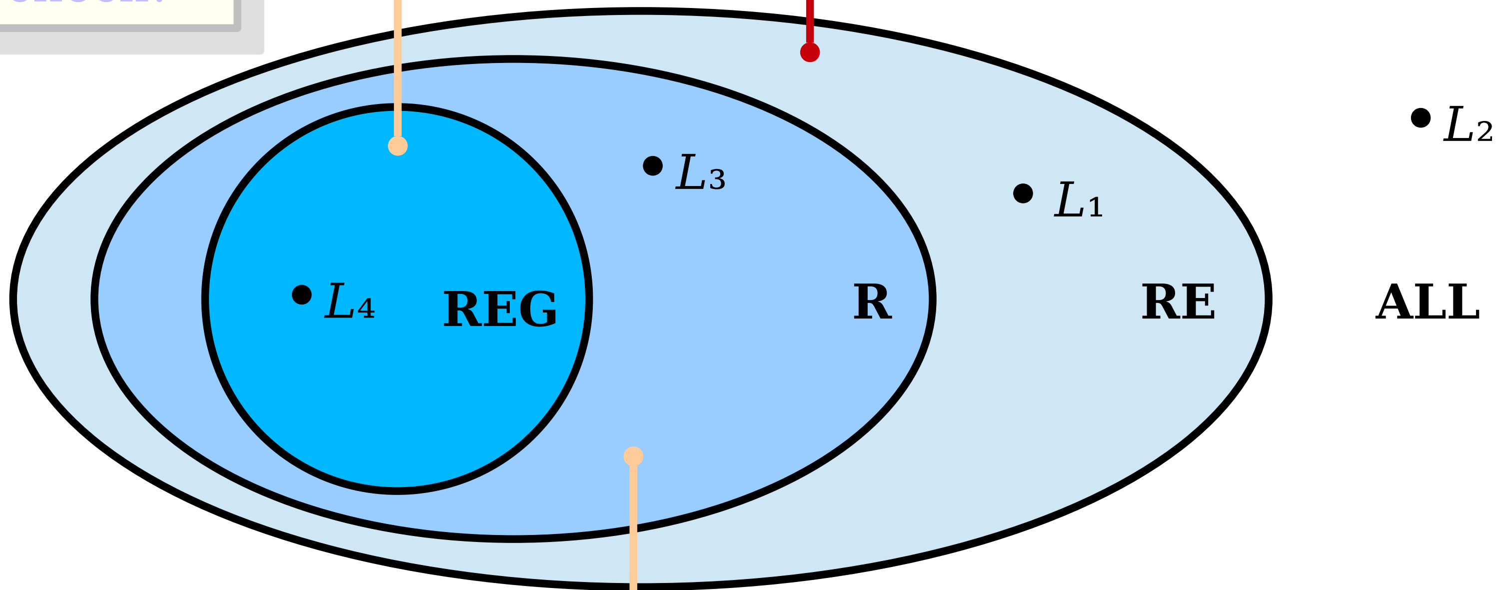
$L_4 = \{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N}$ and $n \leq 1000 \}$

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 $w \notin L$?

First, the **RE** languages. To check whether a language is **RE**, ask yourself whether, for any string in the language, you could prove to someone else that it's in the language.

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

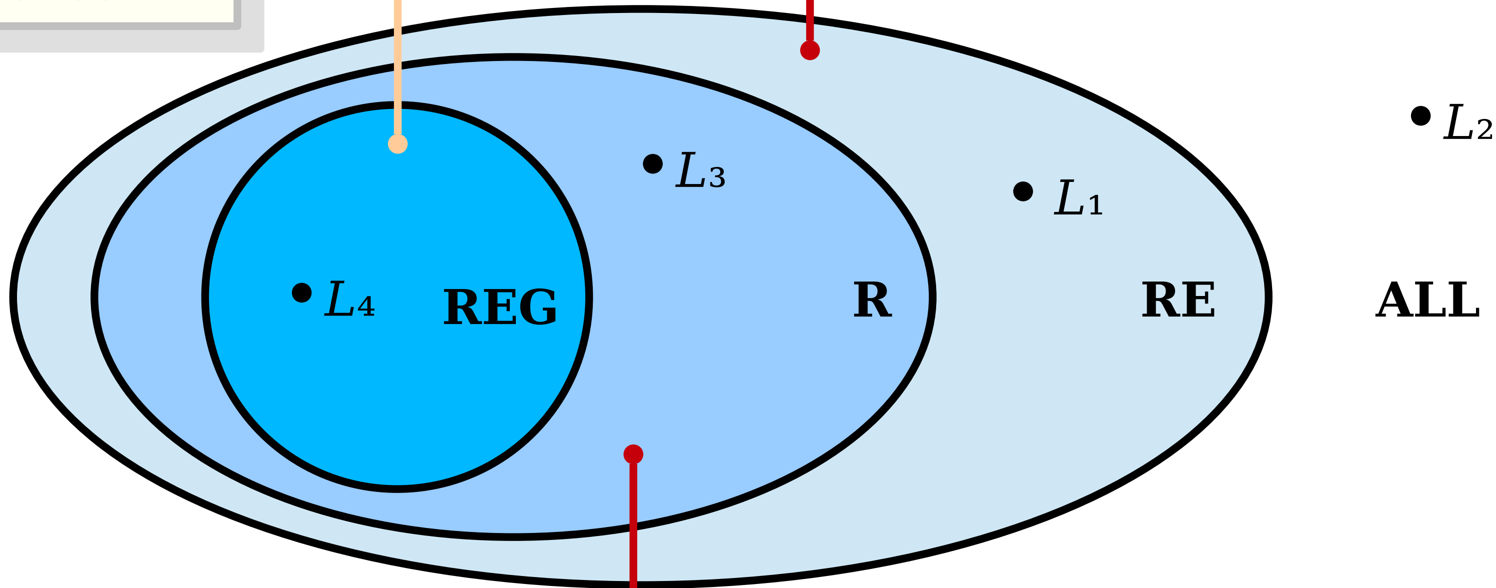
$L_4 = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}$ and $n \leq 1000 \}$

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 $w \notin L$?

Next, the **R** languages. If that you already know your language is in **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.

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

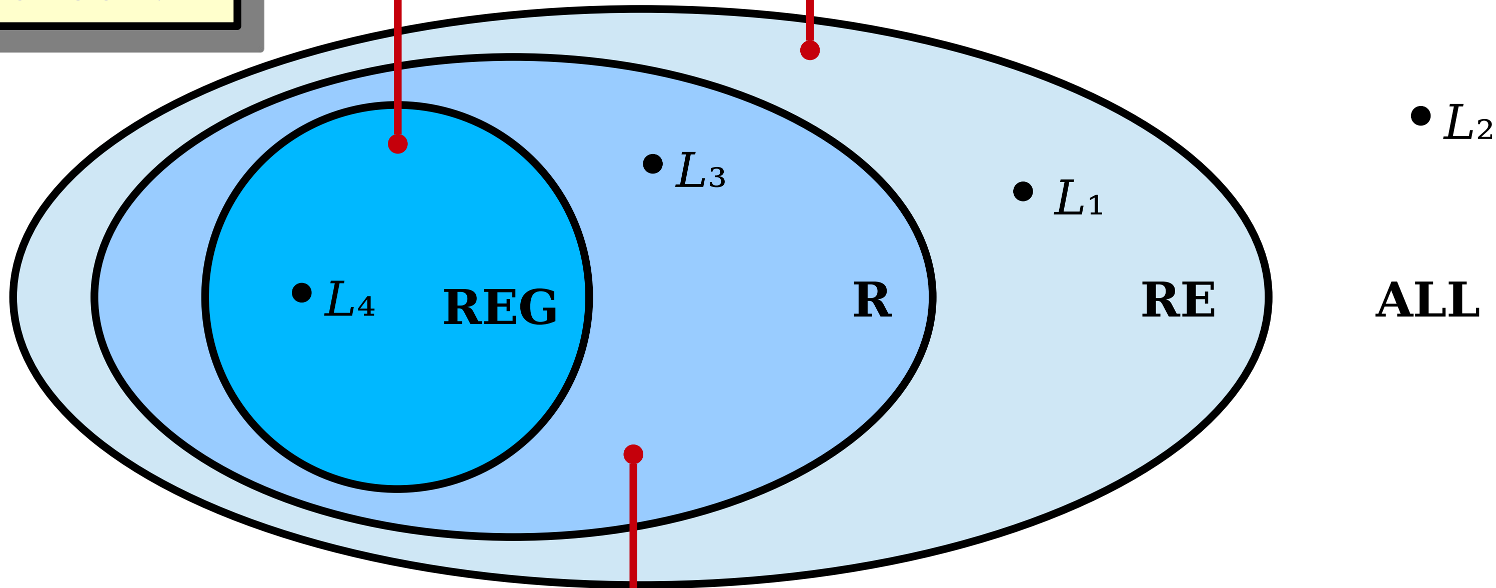
$L_4 = \{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N}$ and $n \leq 1000 \}$

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.

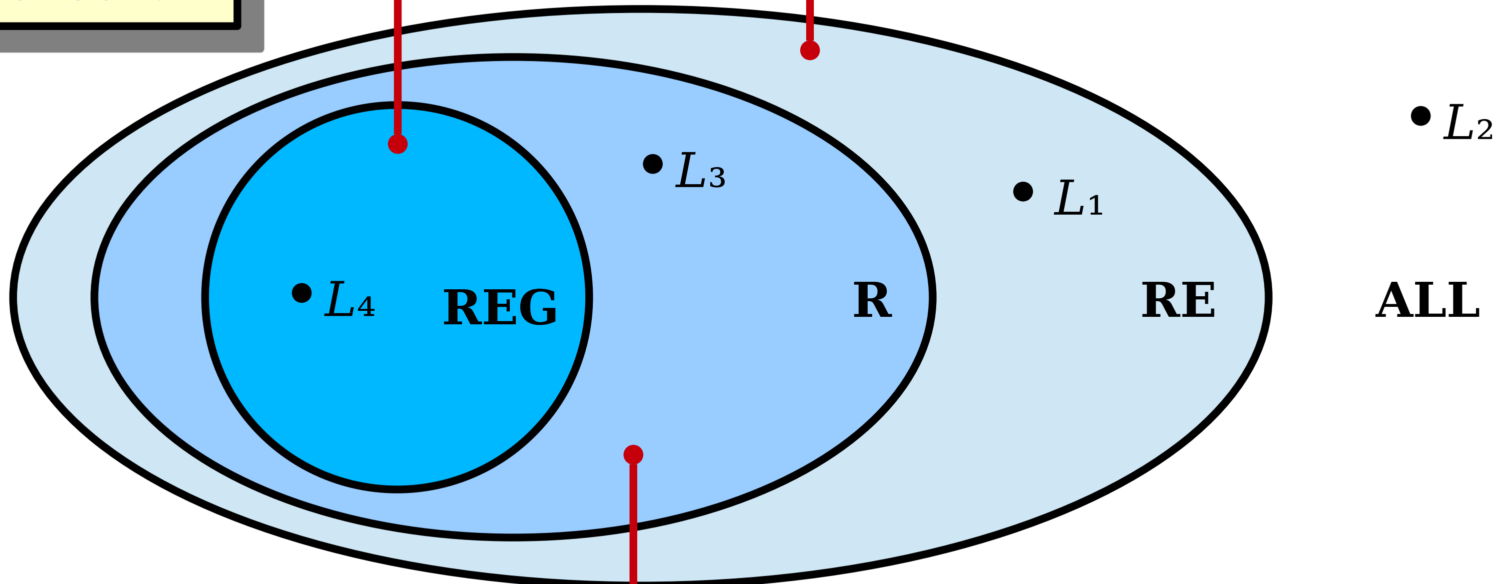
$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$
$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$
$L_3 = \{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N}$ and $n > 1000 \}$
$L_4 = \{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N}$ and $n \leq 1000 \}$

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 $w \in L$?

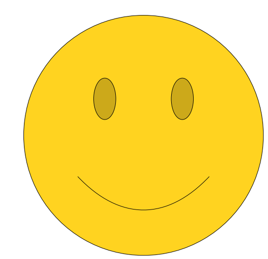


R: Languages with Deciders

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

The more that you learn about these languages, the more intuitions and nuances you'll be able to use to help guide your search.

L_1 = { ⟨M⟩ | M is a TM that accepts at least two strings }
L_2 = { ⟨M⟩ | M is a TM that accepts exactly two strings }
L_3 = { $\mathtt{a}^n\mathtt{b}^n$ | $n \in \mathbb{N}$ and $n > 1000$ }
L_4 = { $\mathtt{a}^n\mathtt{b}^n$ | $n \in \mathbb{N}$ and $n \leq 1000$ }

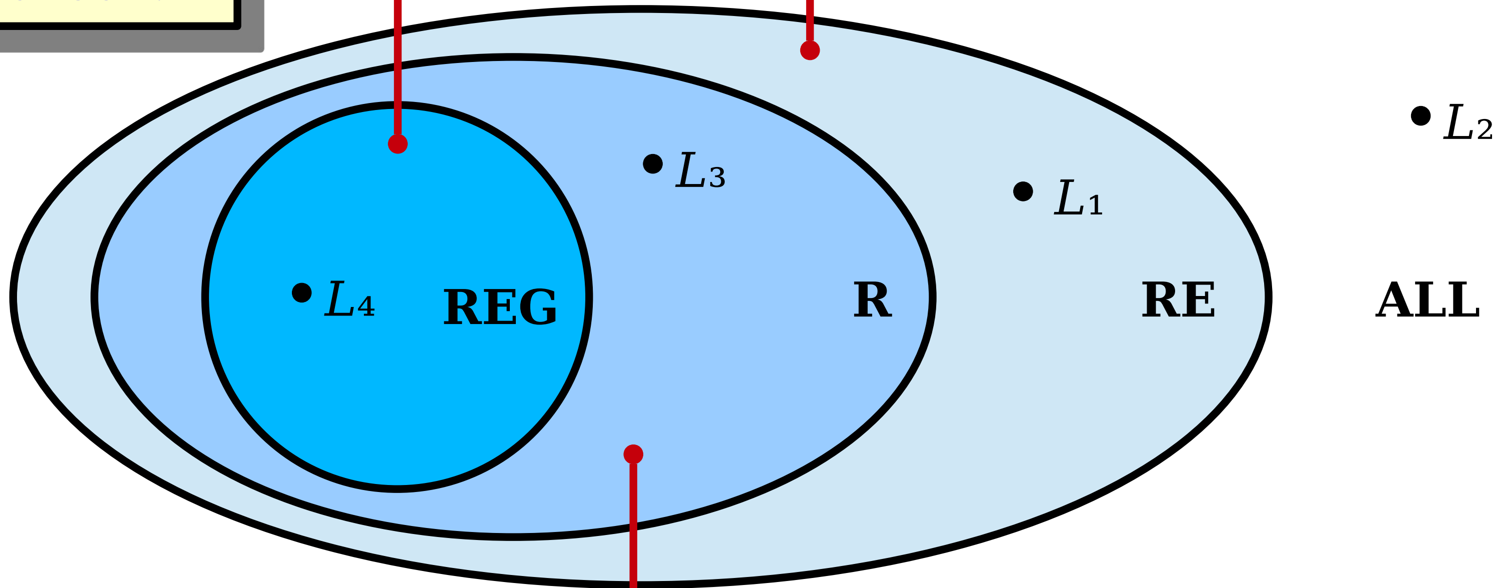


REG: Problems Solvable with Finite Memory

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 $w \notin L$?

Hopefully, this gives you a good starting point for working through Lava Diagram questions. Good luck!

$L_1 = \{ \langle M \rangle \mid M$ is a TM that accepts at least two strings $\}$

$L_2 = \{ \langle M \rangle \mid M$ is a TM that accepts exactly two strings $\}$

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

$L_4 = \{ \texttt{a}^n\texttt{b}^n \mid n \in \mathbb{N}$ 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.