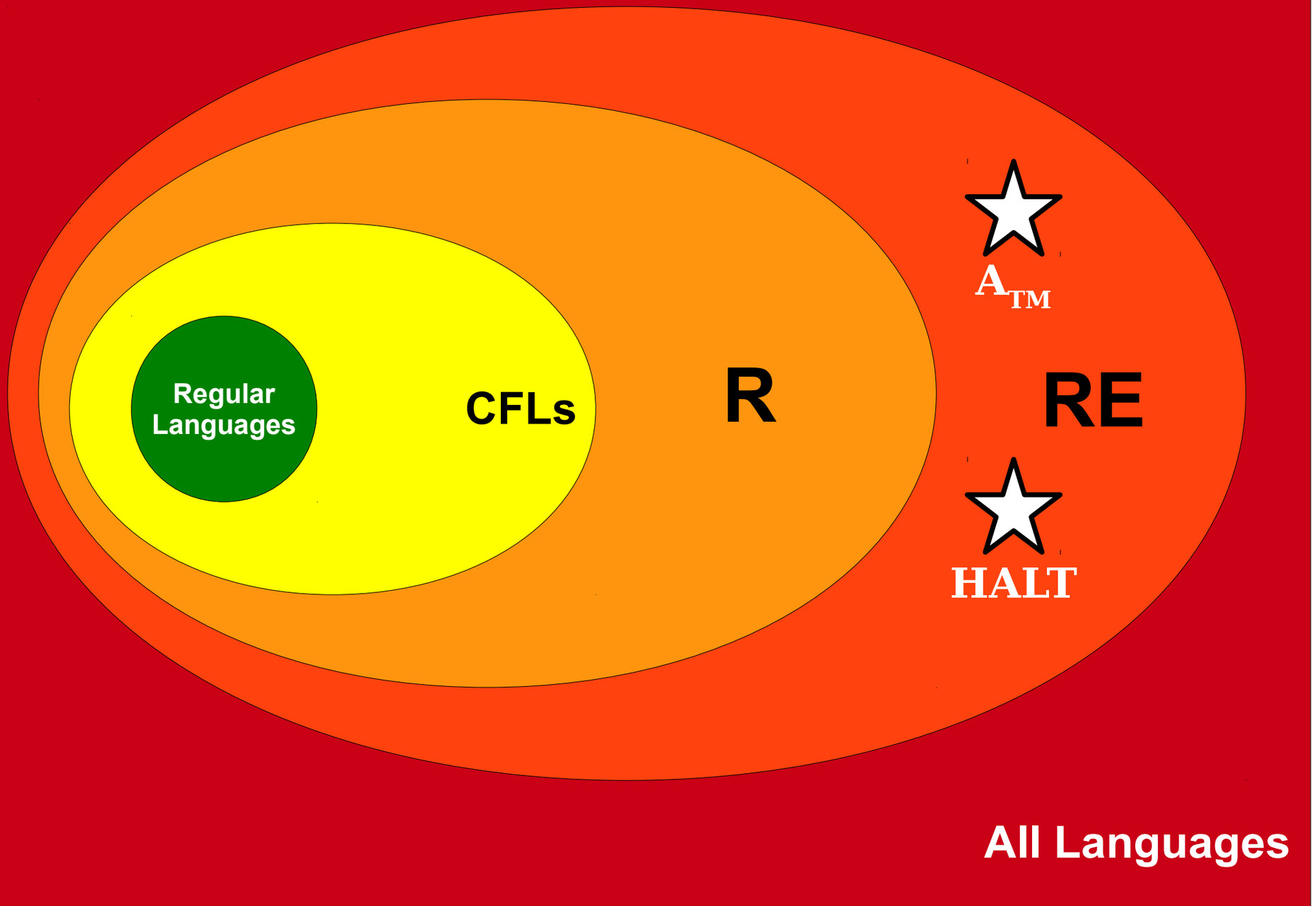
# The Lava Diagram (Wrap-Up of Undecidability)

# RE

(Lava Diagram Problem Tips)




***How do you identify an RE language?***

# RE
## (Lava Diagram Problem Tips)

- You've seen two examples of RE (but not R) problems so far: $A_{TM}$, *HALT.* They both have this form:
  - $L = \{\langle M,w\rangle \mid$ /* something about $M$'s behavior on $w$ */$\}$
- You've also seen how we used the Trickster approach to prove both are undecidable (not in R).
- *You may have found yourself thinking, "Couldn't I always just use Trickster on <u>any</u> language of TMs to show it's undecidable?" And the answer, in fact, is yes! (with a couple caveats)*

# Rice's Theorem
## (Lava Diagram Problem Tips)

- **Any language that consists of strings that are TM code, and <u>filters</u> those TMs based on a criteria that has anything to do with the TM's <u>language or behavior</u>, is undecidable.***
  - * In other words, not in R; at best in RE, maybe not even that.
- Meaning languages matching these sorts of templates will always be undecidable:
  - $L = \{\langle M,w\rangle \mid$ /* something about $M$'s behavior on $w$ */$\}$
  - $L = \{\langle M\rangle \mid$ /* something about $M$'s behavior */$\}$
  - $L = \{\langle M\rangle \mid$ /* something about $M$'s language */$\}$
  - $L = \{\langle M_1, M_2\rangle \mid$ /* something comparing $M_1$ and $M_2$'s languages or behaviors */$\}$
- Example criteria: "$M$ accepts at least one string," "$M$'s language is finite," "$M$ loops on $w$"

# Rice's Theorem
## (Lava Diagram Problem Tips)

- **Any language that consists of strings that are TM code, and filters those TMs based on a criteria that has anything to do with the TM's language or behavior, is undecidable.**
- **Caveat 1:** not all languages of the form $L = \{\langle M \rangle \mid$ /* criteria */$\}$ are undecidable; only when the criteria filters based on the TM's language or behavior.
- Questions relating purely to the *text* of the code, and not its *behavior when run* are generally decidable. Decidable examples:
  - $L_1 = \{\langle M,w \rangle \mid$ the string $\langle M \rangle$ has more a's than the string $w\}$
  - $L_2 = \{\langle M \rangle \mid$ the string $\langle M \rangle$ has odd length$\}$
  - *Intuition: You don't need to try running the TM M to see how many a's its code has or that its code is an odd-length string.*
- **Caveat 2:** The criteria can't be true of zero TMs or all TMs (those languages are just $\emptyset$ and $\Sigma^*$, respectively). Rice's Theorem doesn't apply unless the criteria actually "filters" some in/some out.

# Rice's Theorem
## (Lava Diagram Problem Tips)

- **Any language that consists of strings that are TM code, and filters those TMs based on a criteria that has anything to do with the TM's language or behavior, is undecidable.**
- **Caveat 1:** not all languages of the form $L = \{\langle M \rangle \mid$ /* criteria */$\}$ are undecidable; only when the criteria filters based on the TM's language or behavior.
- Questions relating purely to the *text* of the code, and not its *behavior when run* are generally decidable. Decidable examples:
  - $L_1 = \{\langle M,w \rangle \mid$ the string $\langle M \rangle$ has more a's than the string $w\}$
  - $L_2 = \{\langle M \rangle \mid$ the string $\langle M \rangle$ has odd length$\}$
  - *Intuition: You don't need to try running the TM M to see how many a's its code has or that its code is an odd-length string.*
- **Caveat 2:** The criteria can't be true of zero TMs or all TMs (those languages are just $\emptyset$ and $\Sigma^*$, respectively). Rice's Theorem doesn't apply unless the criteria actually "filters" some in/some out.

**Quick check:**
How would you categorize $L_1$ in the Lava Diagram?

# Rice's Theorem
## (Lava Diagram Problem Tips)

- **Any language that consists of strings that are TM code, and filters those TMs based on a criteria that has anything to do with the TM's language or behavior, is undecidable.**
- **Caveat 1:** not all languages of the form $L = \{\langle M \rangle \mid$ /* criteria */$\}$ are undecidable; only when the criteria filters based on the TM's language or behavior.
- Questions relating purely to the *text* of the code, and not its *behavior when run* are generally decidable. Decidable examples:
  - $L_1 = \{\langle M,w \rangle \mid$ the string $\langle M \rangle$ has more a's than the string $w\}$
  - $L_2 = \{\langle M \rangle \mid$ the string $\langle M \rangle$ has odd length$\}$
  - *Intuition: You don't need to try running the TM M to see how many a's its code has or that its code is an odd-length string.*
- **Caveat 2:** The criteria can't be true of zero TMs or all TMs (those languages are just $\emptyset$ and $\Sigma^*$, respectively). Rice's Theorem doesn't apply unless the criteria actually "filters" some in/some out.

**Quick check:**
How would you categorize $L_2$ in the Lava Diagram?

# Beyond RE

# How Many Problems Are Outside the Reach of Turing Machines?

- It’s sad to think about problems that no TM can even *recognize!* How many are there? Hopefully only a few!
- A TM is fundamentally just a string (a piece of code). Let’s say we can interpret any string as a TM (if not syntactically correct according to some TM code system, we’ll just say it is a TM that always rejects).
- So the number of TMs that exist is $|\Sigma^*|$ (the count of all possible strings).

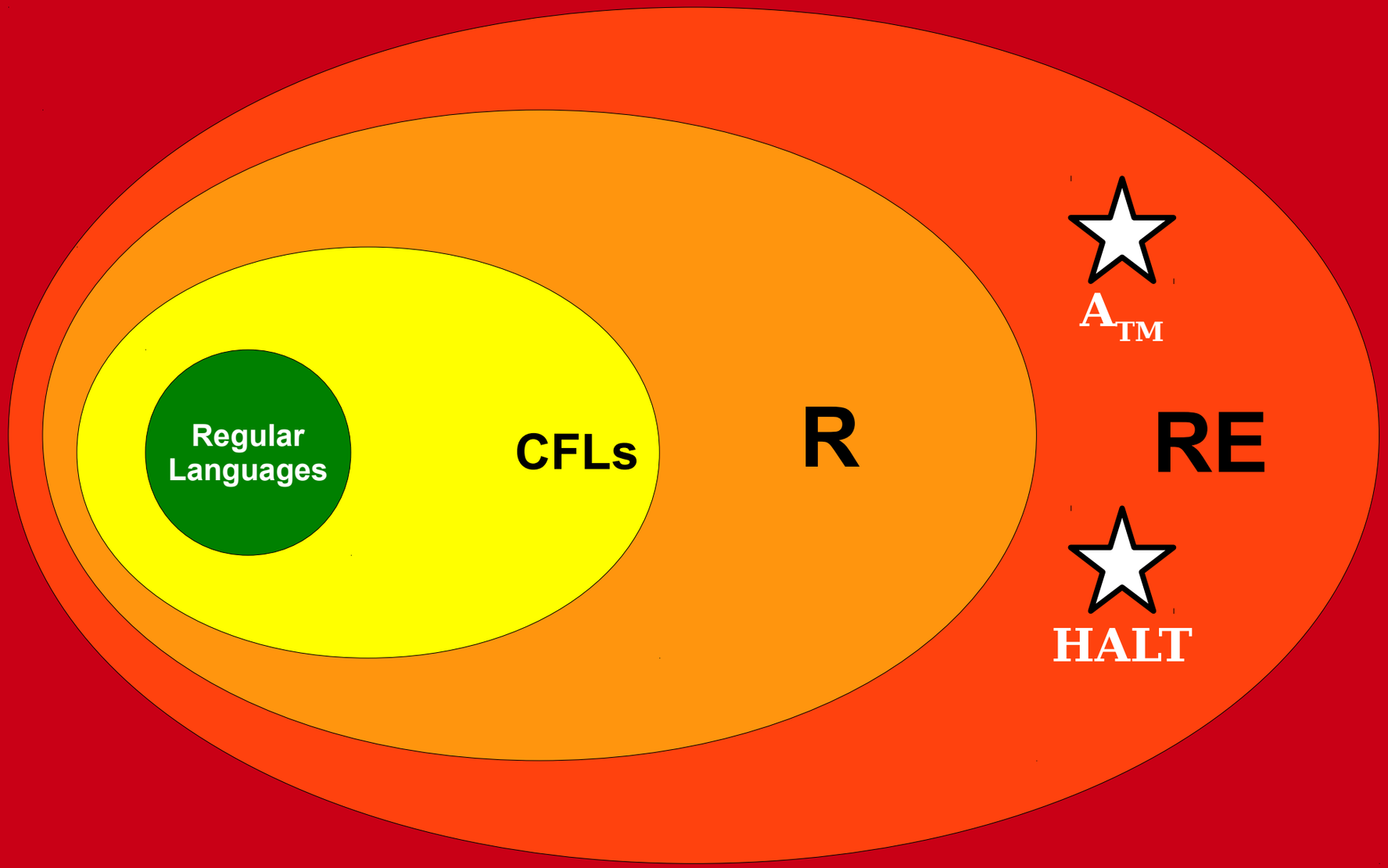
# How Many Problems Are Outside the Reach of Turing Machines?

- It's sad to think about problems that no TM can even *recognize!* How many are there? Hopefully only a few!
- A TM is fundamentally just a string (a piece of code). Let's say we can interpret any string as a TM (if not syntactically correct according to some TM code system, we'll just say it is a TM that always rejects).
- So the number of TMs that exist is $|\Sigma^*|$ (the count of all possible strings).

**Quick check:** How many **languages** are there? Express as the cardinality of some set.

# How Many Problems Are Outside the Reach of Turing Machines?

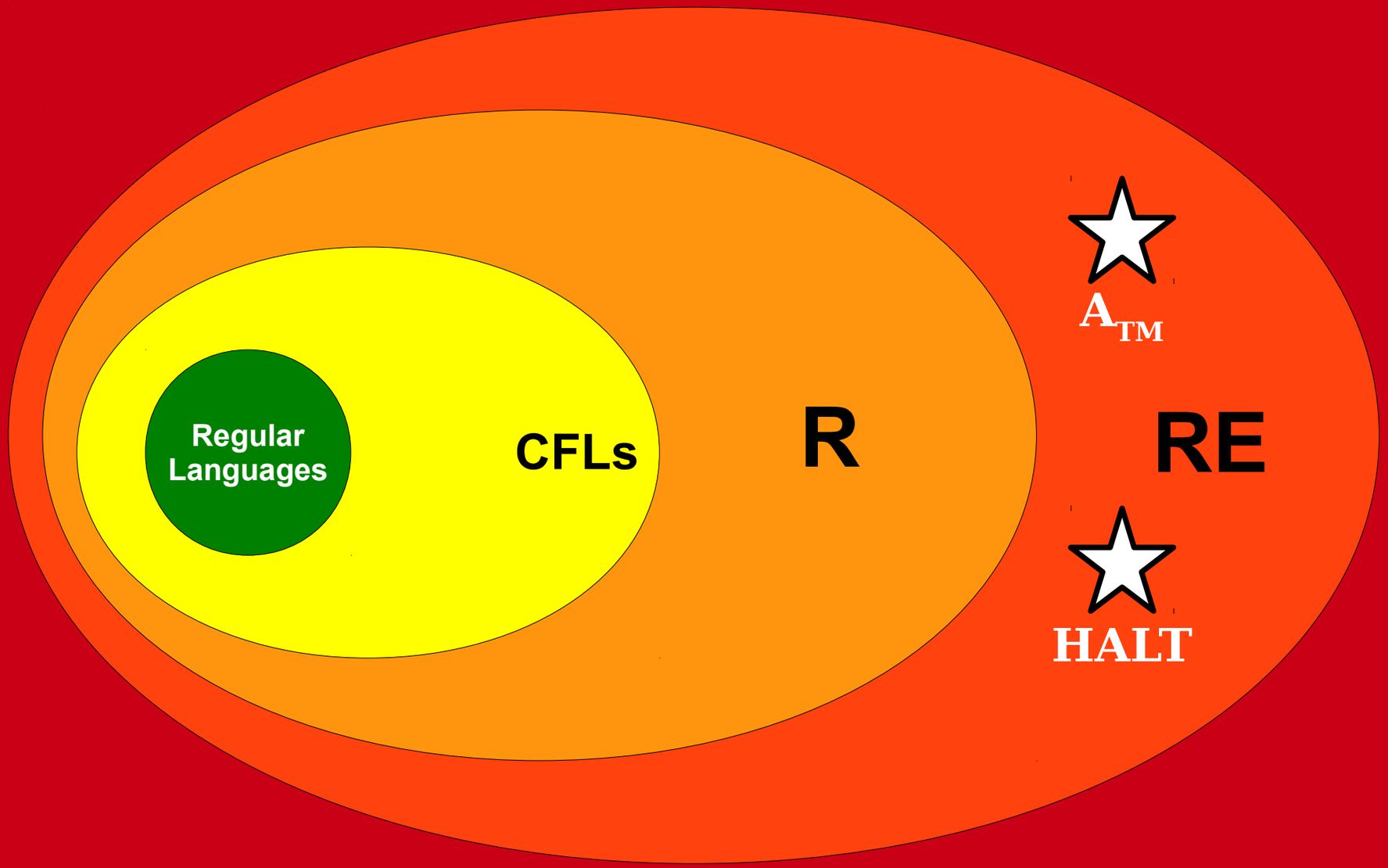
- Sadly, Cantor's Theorem says $|\Sigma^*| < \wp(|\Sigma^*|)$.
- Not nearly enough Turing machines to go around for all the languages that exist.
- So there are many problems (languages) that we can't solve with TMs.




***What is outside RE?***

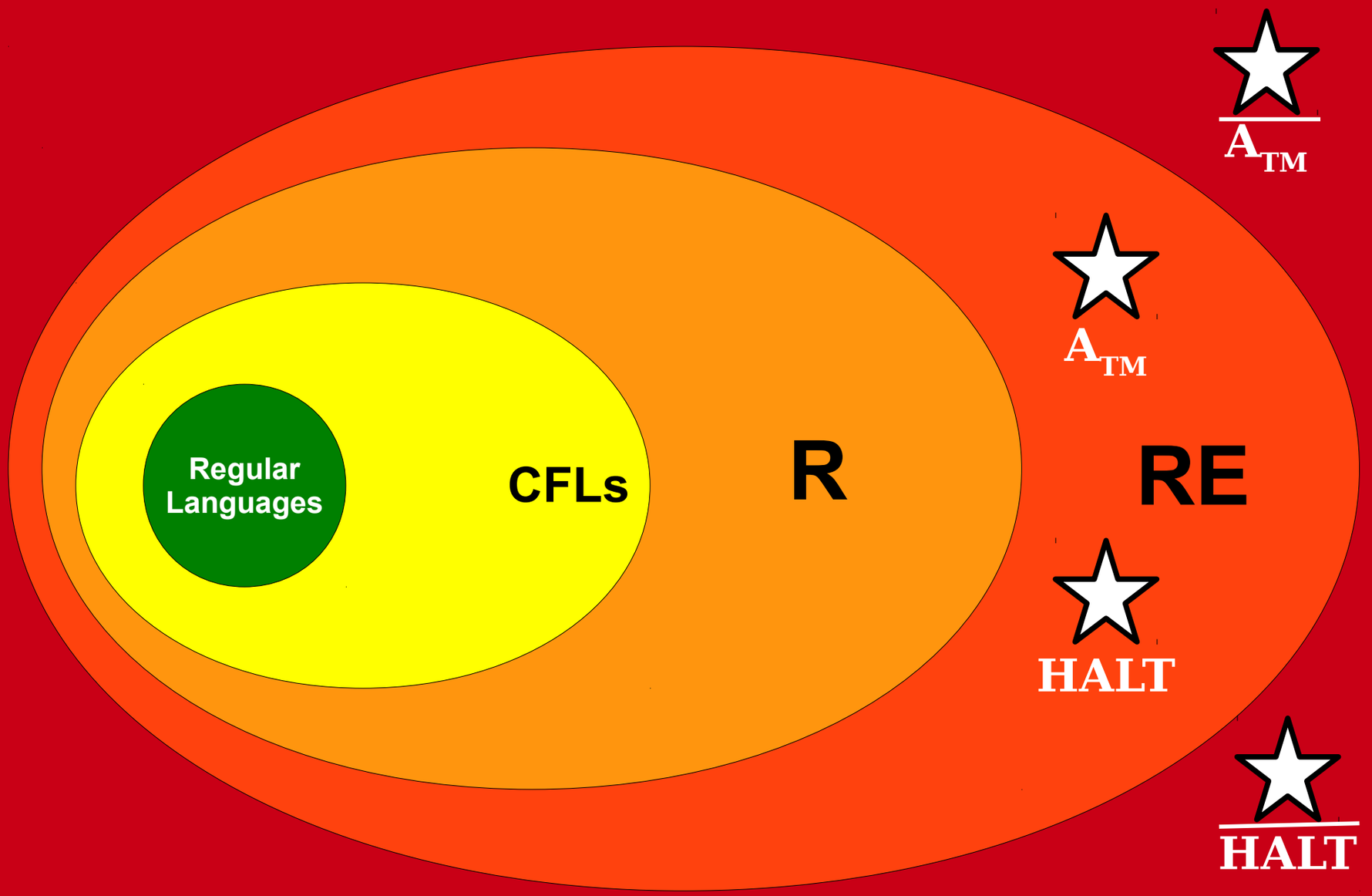
# The Class Unrecognizable
(Lava Diagram Category “All Languages”)

- Languages $L$ that are outside RE are those where:
  - We cannot even build a recognizer TM $M$, where $\mathscr{L}(M) = L$.
  - We cannot build a verifier for $L$.
    - *i.e., there is no finite-length certificate/hint that proves a string w is in L.*
- Fun fact: this class includes all the complements of undecidable RE languages.




# *What makes a language non-RE?*

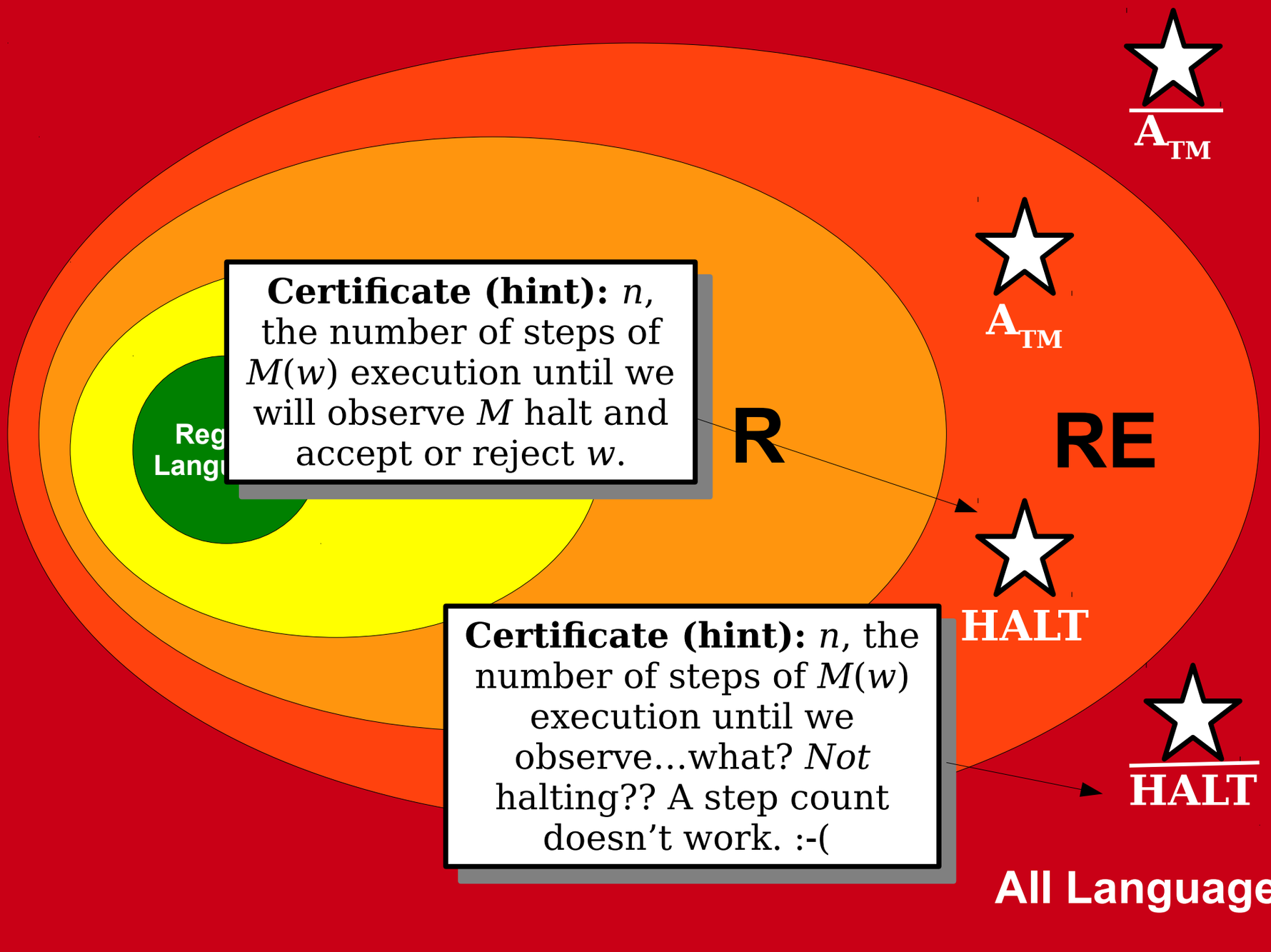
In our discussion of verifiers, we've brushed up against one reason a language could be unverifiable: it's inherently hard to provide a piece of evidence to prove a negative.




# *What makes a language non-RE?*

In our discussion of verifiers, we've brushed up against one reason a language could be unverifiable: it's inherently hard to provide a piece of evidence to prove a negative.

**The complements of undecidable RE languages *are always outside of RE.***




# *What makes a language non-RE?*

In our discussion of verifiers, we've brushed up against one reason a language could be unverifiable: it's inherently hard to provide a piece of evidence to prove a negative.

**The complements of undecidable RE languages *are always outside of RE.***

# The Class Unrecognizable
(Lava Diagram Category “All Languages”)

- Example languages:
  - The complements of undecidable RE languages.
  - $EQ_{TM} = \{\langle M_1, M_2 \rangle \mid \text{where } \mathscr{L}(M_1) = \mathscr{L}(M_1) \}$
  - $\{\langle M \rangle \mid M \text{ loops on at least five strings } \}$

# The Class Unrecognizable
## (Lava Diagram Category “All Languages”)

- Example languages:
  - The complements of undecidable RE languages.
  - $EQ_{TM} = \{ \langle M_1, M_2 \rangle \mid \text{where } \mathscr{L}(M_1) = \mathscr{L}(M_1) \}$
  - $L = \{ \langle M \rangle \mid M \text{ loops on at least five strings} \}$

**Quick check:** A very similar language is: $L' = \{ \langle M \rangle \mid M$ **accepts** at least five strings $\}$. How would you categorize it on the Lava Diagram?

# The Class Unrecognizable
(Lava Diagram Category “All Languages”)

- Example languages:
  - The complements of undecidable RE languages.
  - $EQ_{TM} = \{ \langle M_1, M_2 \rangle \mid \text{where } \mathscr{L}(M_1) = \mathscr{L}(M_1) \}$
  - $\{ \langle M \rangle \mid M \text{ loops on at least five strings} \}$
    - Note that we *can* write a verifier for the language $\{ \langle M \rangle \mid M$ **accepts** at least five strings $\}$, using the certificate $\langle w_1, n_1, w_2, n_2, w_3, n_3, w_4, n_4, w_5, n_5 \rangle$, which is a mouthful, but still finite.

**Quick check:** A very similar language is: $L' = \{ \langle M \rangle \mid M$ **accepts** at least five strings $\}$. How would you categorize it on the Lava Diagram?

# The Language $L_D$

- $L_D = \{\langle M \rangle \mid M \text{ does not accept } \langle M \rangle \}$

  $= \{\langle M \rangle \mid \langle M \rangle \in \mathscr{L}(M)\}$

- This is a classic example of an unrecognizable language.
- To see why we can't build a TM for this language, ask yourself:
  - If you did build a TM for $L_D$, called $M_{LD}$, would $M_{LD}$ accept $\langle M_{LD} \rangle$?

# Happy Story Time

In a certain isolated town, every house has a lawn and the city requires them all to be mowed. The town has only one gardener, who is also a resident of the town, and this gardener mows the lawns of residents iff they do not mow their own lawn.

# Happy Story Time

In a certain isolated town, every house has a lawn and the city requires them all to be mowed. The town has only one gardener, who is also a resident of the town, and this gardener mows the lawns of residents iff they do not mow their own lawn.

**True or false:** The gardener mows their own lawn.

# The Language $L_D$

- $L_D = \{\langle M \rangle \mid M \text{ does not accept } \langle M \rangle \}$

  $= \{\langle M \rangle \mid \langle M \rangle \in \mathscr{L}(M)\}$

- This is a classic example of an unrecognizable language.
- To see why we can't build a TM for this language, ask yourself:
  - If you did build a TM for $L_D$, called $M_{LD}$, would $M_{LD}$ accept $\langle M_{LD} \rangle$?

The existence of a TM for $L_D$ creates a contradiction (paradox), so that's our proof that a TM for $L_D$ cannot exist.