
Rice's Theorem:
"Turing Machines Decide Only Trivia
About Turing Machines"

Robert Keller
April 2009

Functional Properties

- A property of (a machine accepting) a Turing-acceptable language is called **functional** if it only depends on the language accepted, and not on the characteristics of the specific machine that accepts the language.

Example of a Non-Functional Property

- Let $P(\langle M \rangle)$ be the property M halts within 100 steps when started on a blank tape.
- The language of all $\langle M \rangle$ for which $P(\langle M \rangle)$ is true is **non-functional**:

Different machines computing the **same** language may have or not have the property P .

Trivial Properties

- A property P of languages is called "trivial" if either:
 - P holds for all languages, or
 - P holds for no language.

Examples of Non-Trivial Properties of languages L

- L is empty
- L is all strings over the alphabet
- L is regular
- L is context-free
- ...

Rice's Theorem

- The only **decidable** functional properties are the (two) trivial ones.
- In other words, there is no algorithm that will decide of the properties of languages accepted by Turing machines, other than the trivial properties.

Proof of Rice's Theorem

- We'll first do the proof for a special case to make it more concrete, then observe that the method is completely general.
- Suppose we want to show the property of being **regular** is undecidable, i.e. there is no way to test whether an arbitrary Turing machine accepts a regular language.

Proof of Rice

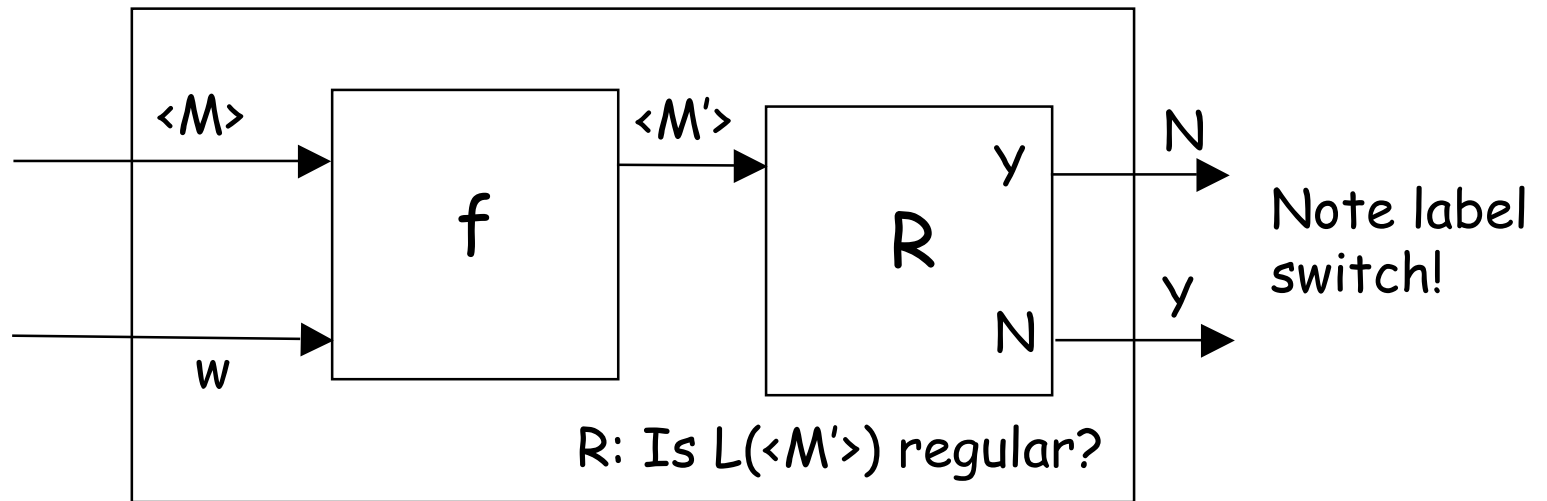
- We have to ask this question first:
 - Does the empty language have the property?
 - For the property of being regular, the answer is ____.
 - If so, let S be some Turing-acceptable language that does **not** have the property. For the question of regularity, we could let S be _____.

Rice Proof, continued

- Let Q be a Turing machine accepting S , the chosen non-regular language.
- Now assume that we can determine whether the language of an arbitrary TM is regular. Let R be a machine that decides this.
- We use R to decide the undecidable.

Rice Template

Does M halt on w ?



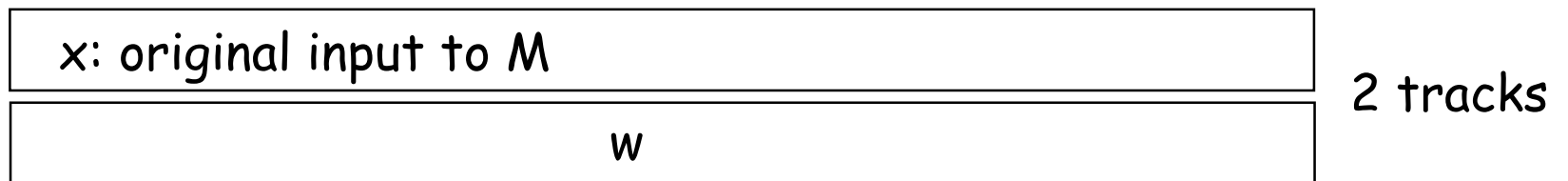
What is f ?

f:

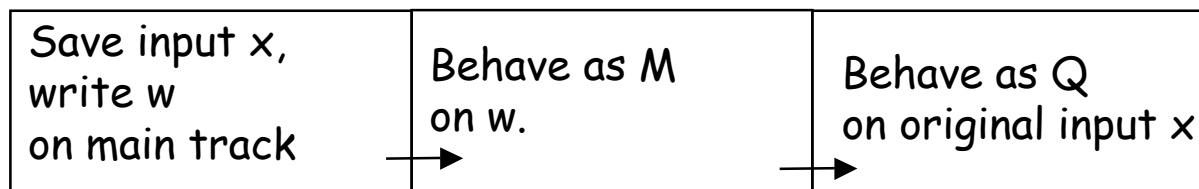
- With input $\langle M \rangle$ and w , f constructs a Turing machine $\langle M' \rangle$ that behaves as follows:
 - M' puts its input x aside (e.g. on a separate tape track) for later recall.
 - M' then writes w on the main tape track and **behaves as M on w** .
 - **If this behavior halts**, then M' next behaves as Q (the machine accepting a non-regular language) on x (the original tape of M' that was set aside).
 - **If this behavior does not halt** (which can't be detected), M' does not halt either.

Picture of M' in action

Tape:



Control:



M'

- The language of M' is exactly one of two possibilities:
 - The language of Q , i.e. S , a non-regular language, if M halts on w .
 - The empty language, i.e. a regular language, if M does not halt on w .

Rice Conclusion

- Once M' is constructed, it is passed to R , the regularity tester.
- If R says that $L(M')$ *is* regular, it must be that $L(M) = \emptyset$, so R is essentially saying that M does not halt on w .
- If R says that $L(M')$ is *not* regular, it is saying that M halts on w .
- So if R exists, we could use it to construct a halt-checker.
- Thus R cannot exist.

The General Case

- We worked with the property of being **regular**. What about other properties?
- For any property P , we must first ask whether the **empty language** has the property P or not.
- If it does, we proceed exactly as before, with S being a computable language **not** having property P .
- If it does not, we let S be a language **having** property P . In this case, we **don't switch the labels** on the outer box in the template.

Aside

- Consider a language L such as:

$$L = \begin{cases} \text{empty, if there is life on Jupiter} \\ \text{all strings in } \{0, 1\}^* \text{ otherwise.} \end{cases}$$

- L is **decidable**, because L is one of two decidable sets. We just don't know which set (as of April 29, 2009).