

Teorema di Rice

Introduction to the Rice Theorem

Ci sono molti teoremi che non possono essere decisi, vedere Halting Theorem and Reducibility. Qui andiamo a chiederci quale sia l'insieme dei problemi decidibili.

Proprietà dei linguaggi TM

Data una macchina \mathcal{M} definiamo il suo linguaggio come

$$L_{\mathcal{M}} = \{x \in \Sigma^* : \mathcal{M} \text{ accetta } x\}$$

Allora con questa definizione di linguaggio possiamo dire che una **proprietà**, ossia una funzione da tutti i TM possibili a $\{0, 1\}$ tale per cui se il linguaggio riconosciuto è lo stesso, ossia

$$L_{\mathcal{M}} = L_{\mathcal{M}'} \implies P(\mathcal{M}) = P(\mathcal{M}')$$

Definiamo questa **non triviale** se esiste una macchina per cui è 0, e una per cui è 1 (ossia non è costante). Practically this definition is useful when we need to have a difference between the language and the Turing machine that decides that language.

Three properties of Turing Machines

1. Language properties (what language does it decide? This property concerns Rice's Theorem)
2. Structural properties (what are constituents of turing machine?)
3. Algorithmic properties (how is computing) It is important to note that only Language properties concerns Rice's Lemma.

Enunciato di Rice

Se P è una proprietà dei linguaggi TM, allora è indecidibile il problema “ \mathcal{M} ha la proprietà P ”.

Proof of Rice We want to prove that the language

$$\{\langle M \rangle : code(\mathcal{M}) \wedge P(M) = 1\}$$

is undecidable. We proved this by Mapping reducibility with the HALT language.

Without loss of generality, we assume that given a P we have that $P(\mathcal{M}_\emptyset) = 0$. Then, given the fact that the property is not trivial we have that exists a \mathcal{M} such that $P(\mathcal{M}) = 1$. Let's procede by contradiction. Assume that P is decidable. Let's proof that $HALT \leq P$ where P is the language that knows the same stuff.

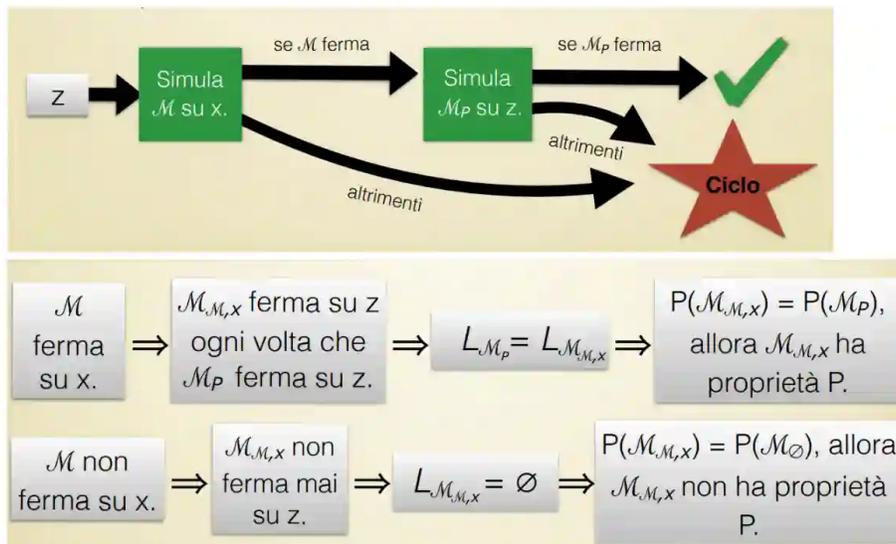


Figure 1: Teorema di Rice-20240313115713766

This proves Rice Theorem by mapping reducibility properties.

L'insieme delle funzioni non decidibili

Qui andiamo a dimostrare che la stragrande maggioranza dei linguaggi non sono riconoscibili. TODO: questo si potrebbe ripassare dalle slides e verrebbe fatto in maniera diversa.

Unione di insiemi numerabili è numerabile Vedi Descrizione linguaggio#Numerabilità per alfabeti per costruzione e dimostrazione. Sarebbe buono saperlo fare da solo. L'idea è avere un parametro che di dice quanto è l'esponente dell'insieme. E poi andare per sorta di ricorsione. ##### L'insieme delle TM è numerabile. Basta vedere che l'insieme delle TM è un sottoinsieme di A^* , che è numerabile. Questo quando usiamo la codifica binaria, quindi $A = \{0, 1\}$. ##### L'insieme dei linguaggi su alfabeto finito non è numerabile

Possiamo rappresentare un linguaggio su un alfabeto con funzioni indicatrici. Avremmo così una stringa binaria che ci indica o no se una stringa è presente nel linguaggio o meno. Allora posso praticamente usare lo stesso argomento usato in diagonalizzazione di Cantor e avere il risultato.