# Esercitazioni II

Docente: Fabio Zanasi, Tutor: Melissa Antonelli

melissa.antonelli2@unibo.it

1 Marzo 2023

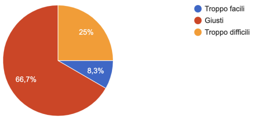
## Prima di iniziare

- Esercizi scaricabili da virtuale e presentati in slide.
- Dal pomeriggio troverete le soluzioni di esercizi svolti e facoltativi su virtuale.
- Per ogni dubbio (anche su esercizi precedenti o facoltativi), scrivetemi: `melissa.antonelli2@unibo.it`

# Prima di iniziare (Piccoli Cambiamenti)

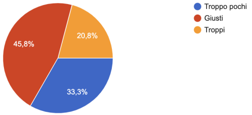
Come valuteresti la difficoltà degli esercizi assegnati durante il l'**esercitazione**?

24 risposte




Come vasteresti il numero degli esercizi assegnati durante l'**esercitazione**?

24 risposte



## Prima di iniziare (Piccoli Cambiamenti)

- Qualche esercizio in piú: 5/6 (+ 4).
- Esercizi di difficoltá crescente (per argomento).
- Avrete un tempo assegnato per svolgere gli esercizi autonomamente (facendo domande).
- Argomenti di oggi: 4 su *mapping reduction* + 2 su Teorema di Rice.
- Correzione (dei primi) collettiva; se volete, svolta da voi.

# Mapping Reduction

**Problema 1.**

Mostra che $\leq$ é una relazione transitiva.

## Correzione tra 15/20 minuti!

**Ricorda che**...

**Definizione (Mapping-Reduction)**

Siano $L$ e $L'$ linguaggi su un dato alfabeto $\Sigma$, diciamo che $L'$ *mapping-riduce* a $L'$, $L \leq L'$, se esiste una funzione computabile $f : \Sigma^* \to \Sigma^*$ tale che:

$$x \in L \quad \text{sse} \quad f(x) \in L'.$$

## Soluzione 1.

Assumi $L \leq L'$ e $L' \leq L''$.

## Soluzione 1.

Assumi $L \leq L'$ e $L' \leq L''$. Per Df. di m-reduciton, esistono due funzioni $f$ e $g$ tali che:

$$\begin{aligned} x \in L \quad & sse \quad f(x) \in L' \\ y \in L' \quad & sse \quad g(y) \in L'' \end{aligned}$$

**Soluzione 1.**

Assumi $L \leq L'$ e $L' \leq L''$. Per Df. di m-reduciton, esistono due funzioni $f$ e $g$ tali che:

$$\begin{array}{lll} x \in L & sse & f(x) \in L' \\ y \in L' & sse & g(y) \in L'' \end{array}$$

Considera la composizione $h(x) = g(f(x))$. Costruisci TM che computa $h$:

## Soluzione 1.

Assumi $L \leq L'$ e $L' \leq L''$. Per Df. di m-reduciton, esistono due funzioni $f$ e $g$ tali che:

$$\begin{aligned} x \in L \quad & sse \quad f(x) \in L' \\ y \in L' \quad & sse \quad g(y) \in L'' \end{aligned}$$

Considera la composizione $h(x) = g(f(x))$. Costruisci TM che computa $h$:

1. Simula TM che computa $f$ su input $x$ e chiama l'output $y$.
2. Simula TM che computa $g$ su $y$.

## Soluzione 1.

Assumi $L \leq L'$ e $L' \leq L''$. Per Df. di m-reduciton, esistono due funzioni $f$ e $g$ tali che:

$$\begin{aligned} x \in L \quad & sse \quad f(x) \in L' \\ y \in L' \quad & sse \quad g(y) \in L'' \end{aligned}$$

Considera la composizione $h(x) = g(f(x))$. Costruisci TM che computa $h$:

1. Simula TM che computa $f$ su input $x$ e chiama l'output $y$.
2. Simula TM che computa $g$ su $y$.

(NB le succitate TM esistono perché le funzioni sono computabili per Df. di m-riducibilitá)

## Soluzione 1.

Assumi $L \leq L'$ e $L' \leq L''$. Per Df. di m-reduciton, esistono due funzioni $f$ e $g$ tali che:

$$\begin{array}{lcl} x \in L & sse & f(x) \in L' \\ y \in L' & sse & g(y) \in L'' \end{array}$$

Considera la composizione $h(x) = g(f(x))$. Costruisci TM che computa $h$:

1. Simula TM che computa $f$ su input $x$ e chiama l'output $y$.
2. Simula TM che computa $g$ su $y$.

(NB le succitate TM esistono perché le funzioni sono computabili per Df. di m-riducibilitá)
L'output é $h(x) = g(f(x))$.

## Soluzione 1.

Assumi $L \leq L'$ e $L' \leq L''$. Per Df. di m-reduciton, esistono due funzioni $f$ e $g$ tali che:

$$x \in L \quad sse \quad f(x) \in L'$$
$$y \in L' \quad sse \quad g(y) \in L''$$

Considera la composizione $h(x) = g(f(x))$. Costruisci TM che computa $h$:

1. Simula TM che computa $f$ su input $x$ e chiama l'output $y$.
2. Simula TM che computa $g$ su $y$.

(NB le succitate TM esistono perché le funzioni sono computabili per Df. di m-riducibilitá)
L'output é $h(x) = g(f(x))$. Dunque $h$ é computabile e

$$x \in L \quad sse \quad h(x) \in L''.$$

Concludiamo che $L \leq L''$ con riduzione $h$.

**Problema 2.**

Dimostra che se $L \leq L'$ e $L'$ é decidibile, allora $L$ é decidibile.

**Correzione tra 15/20 minuti!**

**Ricorda che...**

**Definizione (Linguaggio Decidibile)**

Una TM *decide* un linguaggio $L$ quando: se $x \in L$, allora $M$ accetta $x$; se $x \notin L$, allora $M$ rigetta $x$. Un linguaggio é *decidibile* quando esiste una TM che lo decide.

## Soluzione 2.

Per ipotesi $L'$ é decidibile, dunque esiste TM $M'$ che decide $L'$.

## Soluzione 2.

Per ipotesi $L'$ é decidibile, dunque esiste TM $M'$ che decide $L'$. Per ipotesi $L \leq L'$, esiste $f$ computabile tale che,

$$x \in L \quad \text{sse} \quad f(x) \in L'.$$

## Soluzione 2.

Per ipotesi $L'$ é decidibile, dunque esiste TM $M'$ che decide $L'$. Per ipotesi $L \leq L'$, esiste $f$ computabile tale che,

$$x \in L \quad \text{sse} \quad f(x) \in L'.$$

Consideriamo TM $M$ che, per ogni input $x$:

1. $M$ computa $f(x)$ (computabile per Df. di m-reduction).
2. $M$ esegue $M'$ su input $f(x)$ e restituisce l'output di $M'$.

## Soluzione 2.

Per ipotesi $L'$ é decidibile, dunque esiste TM $M'$ che decide $L'$. Per ipotesi $L \leq L'$, esiste $f$ computabile tale che,

$$x \in L \quad \text{sse} \quad f(x) \in L'.$$

Consideriamo TM $M$ che, per ogni input $x$:

1. $M$ computa $f(x)$ (computabile per Df. di m-reduction).
2. $M$ esegue $M'$ su input $f(x)$ e restituisce l'output di $M'$.

Per Df. di m-reduction, se $x \in L$, allora $f(x) \in L'$.

## Soluzione 2.

Per ipotesi $L'$ é decidibile, dunque esiste TM $M'$ che decide $L'$. Per ipotesi $L \leq L'$, esiste $f$ computabile tale che,

$$x \in L \quad \text{sse} \quad f(x) \in L'.$$

Consideriamo TM $M$ che, per ogni input $x$:

1. $M$ computa $f(x)$ (computabile per Df. di m-reduction).
2. $M$ esegue $M'$ su input $f(x)$ e restituisce l'output di $M'$.

Per Df. di m-reduction, se $x \in L$, allora $f(x) \in L'$. Ma se $f(x) \in L'$, $M'$ deve accettare $f(x)$

## Soluzione 2.

Per ipotesi $L'$ é decidibile, dunque esiste TM $M'$ che decide $L'$. Per ipotesi $L \leq L'$, esiste $f$ computabile tale che,

$$x \in L \quad \text{sse} \quad f(x) \in L'.$$

Consideriamo TM $M$ che, per ogni input $x$:

1. $M$ computa $f(x)$ (computabile per Df. di m-reduction).
2. $M$ esegue $M'$ su input $f(x)$ e restituisce l'output di $M'$.

Per Df. di m-reduction, se $x \in L$, allora $f(x) \in L'$. Ma se $f(x) \in L'$, $M'$ deve accettare $f(x)$ e conseguentemente $M$ accetta $x$.

## Soluzione 2.

Per ipotesi $L'$ é decidibile, dunque esiste TM $M'$ che decide $L'$. Per ipotesi $L \leq L'$, esiste $f$ computabile tale che,

$$x \in L \quad \text{sse} \quad f(x) \in L'.$$

Consideriamo TM $M$ che, per ogni input $x$:

1. $M$ computa $f(x)$ (computabile per Df. di m-reduction).
2. $M$ esegue $M'$ su input $f(x)$ e restituisce l'output di $M'$.

Per Df. di m-reduction, se $x \in L$, allora $f(x) \in L'$. Ma se $f(x) \in L'$, $M'$ deve accettare $f(x)$ e conseguentemente $M$ accetta $x$.
Analogamente, se $x \notin L$, allora $f(x) \notin L'$.

## Soluzione 2.

Per ipotesi $L'$ é decidibile, dunque esiste TM $M'$ che decide $L'$. Per ipotesi $L \leq L'$, esiste $f$ computabile tale che,

$$x \in L \quad \text{sse} \quad f(x) \in L'.$$

Consideriamo TM $M$ che, per ogni input $x$:

1. $M$ computa $f(x)$ (computabile per Df. di m-reduction).
2. $M$ esegue $M'$ su input $f(x)$ e restituisce l'output di $M'$.

Per Df. di m-reduction, se $x \in L$, allora $f(x) \in L'$. Ma se $f(x) \in L'$, $M'$ deve accettare $f(x)$ e conseguentemente $M$ accetta $x$.
Analogamente, se $x \notin L$, allora $f(x) \notin L'$. Se $f(x) \notin L'$, $M'$ rigetta $f(x)$

## Soluzione 2.

Per ipotesi $L'$ é decidibile, dunque esiste TM $M'$ che decide $L'$. Per ipotesi $L \leq L'$, esiste $f$ computabile tale che,

$$x \in L \quad \text{sse} \quad f(x) \in L'.$$

Consideriamo TM $M$ che, per ogni input $x$:

1. $M$ computa $f(x)$ (computabile per Df. di m-reduction).
2. $M$ esegue $M'$ su input $f(x)$ e restituisce l'output di $M'$.

Per Df. di m-reduction, se $x \in L$, allora $f(x) \in L'$. Ma se $f(x) \in L'$, $M'$ deve accettare $f(x)$ e conseguentemente $M$ accetta $x$.
Analogamente, se $x \notin L$, allora $f(x) \notin L'$. Se $f(x) \notin L'$, $M'$ rigetta $f(x)$ e conseguentemente $M$ rigetta $x$.

## Soluzione 2.

Per ipotesi $L'$ é decidibile, dunque esiste TM $M'$ che decide $L'$. Per ipotesi $L \leq L'$, esiste $f$ computabile tale che,

$$x \in L \quad \text{sse} \quad f(x) \in L'.$$

Consideriamo TM $M$ che, per ogni input $x$:

1. $M$ computa $f(x)$ (computabile per Df. di m-reduction).
2. $M$ esegue $M'$ su input $f(x)$ e restituisce l'output di $M'$.

Per Df. di m-reduction, se $x \in L$, allora $f(x) \in L'$. Ma se $f(x) \in L'$, $M'$ deve accettare $f(x)$ e conseguentemente $M$ accetta $x$.
Analogamente, se $x \notin L$, allora $f(x) \notin L'$. Se $f(x) \notin L'$, $M'$ rigetta $f(x)$ e conseguentemente $M$ rigetta $x$.
Ma allora $M$ decide $L$.

## Soluzione 2.

Per ipotesi $L'$ é decidibile, dunque esiste TM $M'$ che decide $L'$. Per ipotesi $L \leq L'$, esiste $f$ computabile tale che,

$$x \in L \quad \text{sse} \quad f(x) \in L'.$$

Consideriamo TM $M$ che, per ogni input $x$:

1. $M$ computa $f(x)$ (computabile per Df. di m-reduction).
2. $M$ esegue $M'$ su input $f(x)$ e restituisce l'output di $M'$.

Per Df. di m-reduction, se $x \in L$, allora $f(x) \in L'$. Ma se $f(x) \in L'$, $M'$ deve accettare $f(x)$ e conseguentemente $M$ accetta $x$.
Analogamente, se $x \notin L$, allora $f(x) \notin L'$. Se $f(x) \notin L'$, $M'$ rigetta $f(x)$ e conseguentemente $M$ rigetta $x$.
Ma allora $M$ decide $L$. Quindi, per Df. di linguaggio decidibile, $L$ é decidibile.

**Problema 2 bis.**

Dimostra che se $L \leq L'$ e $L'$ é riconoscibile, allora anche $L$ é riconoscibile.

**Correzione tra 10/15 minuti!**

**Ricorda che**...

**Definizione (Linguaggio Riconoscibile)**

Una TM $M$ *riconosce* un linguaggio $L$: se $x \in L$, allora $M$ termina; se $x \notin L$, allora $M$ non termina. Un linguaggio é *riconoscibile* quando esiste una TM che lo riconosce.

## Soluzione 2 bis.

Per ipotesi $L'$ é riconoscibile, dunque esiste TM $M'$ che riconosce $L'$.

## Soluzione 2 bis.

Per ipotesi $L'$ é riconoscibile, dunque esiste TM $M'$ che riconosce $L'$. Per ipotesi $L \leq L'$, esiste $f$ computabile tale che,

$$x \in L \quad \text{sse} \quad f(x) \in L'.$$

## Soluzione 2 bis.

Per ipotesi $L'$ é riconoscibile, dunque esiste TM $M'$ che riconosce $L'$. Per ipotesi $L \leq L'$, esiste $f$ computabile tale che,

$$x \in L \quad \text{sse} \quad f(x) \in L'.$$

Consideriamo TM $M$ che, per ogni input $x$:

1. $M$ computa $f(x)$ (computabile per Df. di m-reduction).
2. $M$ esegue $M'$ su input $f(x)$ e se $M'$ termina, $M$ termina; altrimenti, $M$ non termina.

## Soluzione 2 bis.

Per ipotesi $L'$ é riconoscibile, dunque esiste TM $M'$ che riconosce $L'$. Per ipotesi $L \leq L'$, esiste $f$ computabile tale che,

$$x \in L \quad \text{sse} \quad f(x) \in L'.$$

Consideriamo TM $M$ che, per ogni input $x$:

1. $M$ computa $f(x)$ (computabile per Df. di m-reduction).
2. $M$ esegue $M'$ su input $f(x)$ e se $M'$ termina, $M$ termina; altrimenti, $M$ non termina.

Per Df. di m-reduction, se $x \in L$, allora $f(x) \in L'$.

## Soluzione 2 bis.

Per ipotesi $L'$ é riconoscibile, dunque esiste TM $M'$ che riconosce $L'$. Per ipotesi $L \leq L'$, esiste $f$ computabile tale che,

$$x \in L \quad \text{sse} \quad f(x) \in L'.$$

Consideriamo TM $M$ che, per ogni input $x$:

1. $M$ computa $f(x)$ (computabile per Df. di m-reduction).
2. $M$ esegue $M'$ su input $f(x)$ e se $M'$ termina, $M$ termina; altrimenti, $M$ non termina.

Per Df. di m-reduction, se $x \in L$, allora $f(x) \in L'$. Ma se $f(x) \in L'$, $M'$ termina (su $f(x)$)

## Soluzione 2 bis.

Per ipotesi $L'$ é riconoscibile, dunque esiste TM $M'$ che riconosce $L'$. Per ipotesi $L \leq L'$, esiste $f$ computabile tale che,

$$x \in L \quad \text{sse} \quad f(x) \in L'.$$

Consideriamo TM $M$ che, per ogni input $x$:

1. $M$ computa $f(x)$ (computabile per Df. di m-reduction).
2. $M$ esegue $M'$ su input $f(x)$ e se $M'$ termina, $M$ termina; altrimenti, $M$ non termina.

Per Df. di m-reduction, se $x \in L$, allora $f(x) \in L'$. Ma se $f(x) \in L'$, $M'$ termina (su $f(x)$) e conseguentemente $M$ termina (su $x$).

## Soluzione 2 bis.

Per ipotesi $L'$ é riconoscibile, dunque esiste TM $M'$ che riconosce $L'$. Per ipotesi $L \leq L'$, esiste $f$ computabile tale che,

$$x \in L \quad \text{sse} \quad f(x) \in L'.$$

Consideriamo TM $M$ che, per ogni input $x$:

1. $M$ computa $f(x)$ (computabile per Df. di m-reduction).
2. $M$ esegue $M'$ su input $f(x)$ e se $M'$ termina, $M$ termina; altrimenti, $M$ non termina.

Per Df. di m-reduction, se $x \in L$, allora $f(x) \in L'$. Ma se $f(x) \in L'$, $M'$ termina (su $f(x)$) e conseguentemente $M$ termina (su $x$).
Se $x \notin L$, allora $f(x) \notin L'$,

## Soluzione 2 bis.

Per ipotesi $L'$ é riconoscibile, dunque esiste TM $M'$ che riconosce $L'$. Per ipotesi $L \leq L'$, esiste $f$ computabile tale che,

$$x \in L \quad \text{sse} \quad f(x) \in L'.$$

Consideriamo TM $M$ che, per ogni input $x$:

1. $M$ computa $f(x)$ (computabile per Df. di m-reduction).
2. $M$ esegue $M'$ su input $f(x)$ e se $M'$ termina, $M$ termina; altrimenti, $M$ non termina.

Per Df. di m-reduction, se $x \in L$, allora $f(x) \in L'$. Ma se $f(x) \in L'$, $M'$ termina (su $f(x)$) e conseguentemente $M$ termina (su $x$).
Se $x \notin L$, allora $f(x) \notin L'$, se $f(x) \notin L'$, $M'$ non termina (su $f(x)$)

## Soluzione 2 bis.

Per ipotesi $L'$ é riconoscibile, dunque esiste TM $M'$ che riconosce $L'$. Per ipotesi $L \leq L'$, esiste $f$ computabile tale che,

$$x \in L \quad \text{sse} \quad f(x) \in L'.$$

Consideriamo TM $M$ che, per ogni input $x$:

1. $M$ computa $f(x)$ (computabile per Df. di m-reduction).
2. $M$ esegue $M'$ su input $f(x)$ e se $M'$ termina, $M$ termina; altrimenti, $M$ non termina.

Per Df. di m-reduction, se $x \in L$, allora $f(x) \in L'$. Ma se $f(x) \in L'$, $M'$ termina (su $f(x)$) e conseguentemente $M$ termina (su $x$).
Se $x \notin L$, allora $f(x) \notin L'$, se $f(x) \notin L'$, $M'$ non termina (su $f(x)$) e conseguentemente $M$ non termina (su $x$).

## Soluzione 2 bis.

Per ipotesi $L'$ é riconoscibile, dunque esiste TM $M'$ che riconosce $L'$. Per ipotesi $L \leq L'$, esiste $f$ computabile tale che,

$$x \in L \quad \text{sse} \quad f(x) \in L'.$$

Consideriamo TM $M$ che, per ogni input $x$:

1. $M$ computa $f(x)$ (computabile per Df. di m-reduction).
2. $M$ esegue $M'$ su input $f(x)$ e se $M'$ termina, $M$ termina; altrimenti, $M$ non termina.

Per Df. di m-reduction, se $x \in L$, allora $f(x) \in L'$. Ma se $f(x) \in L'$, $M'$ termina (su $f(x)$) e conseguentemente $M$ termina (su $x$).
Se $x \notin L$, allora $f(x) \notin L'$, se $f(x) \notin L'$, $M'$ non termina (su $f(x)$) e conseguentemente $M$ non termina (su $x$).
Ma allora $M$ riconosce $L$.

## Soluzione 2 bis.

Per ipotesi $L'$ é riconoscibile, dunque esiste TM $M'$ che riconosce $L'$. Per ipotesi $L \leq L'$, esiste $f$ computabile tale che,

$$x \in L \quad \text{sse} \quad f(x) \in L'.$$

Consideriamo TM $M$ che, per ogni input $x$:

1. $M$ computa $f(x)$ (computabile per Df. di m-reduction).
2. $M$ esegue $M'$ su input $f(x)$ e se $M'$ termina, $M$ termina; altrimenti, $M$ non termina.

Per Df. di m-reduction, se $x \in L$, allora $f(x) \in L'$. Ma se $f(x) \in L'$, $M'$ termina (su $f(x)$) e conseguentemente $M$ termina (su $x$).
Se $x \notin L$, allora $f(x) \notin L'$, se $f(x) \notin L'$, $M'$ non termina (su $f(x)$) e conseguentemente $M$ non termina (su $x$).
Ma allora $M$ riconosce $L$. Quindi, per Df. di linguaggio riconoscibile, $L$ é riconoscibile.

## Soluzione 2 bis.

Per ipotesi $L'$ é riconoscibile, dunque esiste TM $M'$ che riconosce $L'$. Per ipotesi $L \leq L'$, esiste $f$ computabile tale che,

$$x \in L \quad \text{sse} \quad f(x) \in L'.$$

Consideriamo TM $M$ che, per ogni input $x$:

1. $M$ computa $f(x)$ (computabile per Df. di m-reduction).
2. $M$ esegue $M'$ su input $f(x)$ e se $M'$ termina, $M$ termina; altrimenti, $M$ non termina.

Per Df. di m-reduction, se $x \in L$, allora $f(x) \in L'$. Ma se $f(x) \in L'$, $M'$ termina (su $f(x)$) e conseguentemente $M$ termina (su $x$).
Se $x \notin L$, allora $f(x) \notin L'$, se $f(x) \notin L'$, $M'$ non termina (su $f(x)$) e conseguentemente $M$ non termina (su $x$).
Ma allora $M$ riconosce $L$. Quindi, per Df. di linguaggio riconoscibile, $L$ é riconoscibile.

## Breve Pausa di $5/10$ minuti

**Problema 3.**

Considera il linguaggio

$$U = \{y \in \{0,1\}^* \mid y = \text{code}(M) \ \& \ M \text{ termina su } \texttt{111}\}$$

Dimostra che $U$ é indecidibile sfruttando l'indecidibilitá di $HALT$.

## Problema 3.

Considera il linguaggio

$$U = \{y \in \{\mathtt{0}, \mathtt{1}\}^* \mid y = \text{code}(M) \text{ \& } M \text{ termina su } \mathtt{111}\}$$

Dimostra che $U$ é indecidibile sfruttando l'indecidibilitá di $HALT$.

## Suggerimento

Ricorda che, per il Corollario 1 (Lezione 7), se $L \leq L'$ e $L$ é indecidibile, allora $L'$ é indecidibile.

## Correzione tra 20/25 minuti!

**Soluzione 3.**

Sappiamo che *HALT* é indecidibile.

## Soluzione 3.

Sappiamo che $HALT$ é indecidibile. Per il Cor. 1, per dimostrare l'indecidibilitá di $U$ basta dimostrare $HALT \leq U$.

## Soluzione 3.

Sappiamo che *HALT* é indecidibile. Per il Cor. 1, per dimostrare l'indecidibilitá di $U$ basta dimostrare $HALT \leq U$. Per Df. di *mapping-reduction* questo corrisponde a mostrare che esiste $f$ computabile tale che:

$$\langle y, x\rangle \in HALT \quad \text{sse} \quad f(\langle y, x\rangle) \in U.$$

## Soluzione 3.

Sappiamo che *HALT* é indecidibile. Per il Cor. 1, per dimostrare l'indecidibilitá di $U$ basta dimostrare $HALT \leq U$. Per Df. di *mapping-reduction* questo corrisponde a mostrare che esiste $f$ computabile tale che:

$$\langle y, x \rangle \in HALT \quad \text{sse} \quad f(\langle y, x \rangle) \in U.$$

Costruiamo $f$ come segue:

## Soluzione 3.

Sappiamo che *HALT* é indecidibile. Per il Cor. 1, per dimostrare l'indecidibilitá di $U$ basta dimostrare $HALT \leq U$. Per Df. di *mapping-reduction* questo corrisponde a mostrare che esiste $f$ computabile tale che:

$$\langle y, x\rangle \in HALT \quad \text{sse} \quad f(\langle y, x\rangle) \in U.$$

Costruiamo $f$ come segue:

1. $y \neq \text{code}(M)$ ($y$ non codifica alcuna TM), allora $f(\langle y, x\rangle) = y$.
2. $y = \text{code}(M)$, definiamo TM $M_{M,x}$ tale che:
    - su input `111`, cancella il nastro, $M_{M,x}$ scrive $x$ e simula $M$ su $x$

## Soluzione 3.

Sappiamo che $HALT$ é indecidibile. Per il Cor. 1, per dimostrare l'indecidibilitá di $U$ basta dimostrare $HALT \leq U$. Per Df. di *mapping-reduction* questo corrisponde a mostrare che esiste $f$ computabile tale che:

$$\langle y, x\rangle \in HALT \quad \text{sse} \quad f(\langle y, x\rangle) \in U.$$

Costruiamo $f$ come segue:

1. $y \neq \text{code}(M)$ ($y$ non codifica alcuna TM), allora $f(\langle y, x\rangle) = y$.
2. $y = \text{code}(M)$, definiamo TM $M_{M,x}$ tale che:
   - su input `111`, cancella il nastro, $M_{M,x}$ scrive $x$ e simula $M$ su $x$
   - su ogni altro input, $M_{M,x}$ entra in loop.

## Soluzione 3.

Sappiamo che $HALT$ é indecidibile. Per il Cor. 1, per dimostrare l'indecidibilitá di $U$ basta dimostrare $HALT \leq U$. Per Df. di *mapping-reduction* questo corrisponde a mostrare che esiste $f$ computabile tale che:

$$\langle y, x\rangle \in HALT \quad \text{sse} \quad f(\langle y, x\rangle) \in U.$$

Costruiamo $f$ come segue:

1. $y \neq \text{code}(M)$ ($y$ non codifica alcuna TM), allora $f(\langle y, x\rangle) = y$.
2. $y = \text{code}(M)$, definiamo TM $M_{M,x}$ tale che:
   - su input `111`, cancella il nastro, $M_{M,x}$ scrive $x$ e simula $M$ su $x$
   - su ogni altro input, $M_{M,x}$ entra in loop.

   Definiamo $f(\langle y, x\rangle) = \text{code}(M_{M,x})$.

## Soluzione 3 (Continua)

Dimostriamo che $f$ é computabile e

$$\langle y, x\rangle \in HALT \quad \text{sse } f(\langle y, x\rangle) \in U$$

ovvero $HALT \leq U$.

## Soluzione 3 (Continua)

Dimostriamo che $f$ é computabile e

$$\langle y, x\rangle \in HALT \quad \text{sse } f(\langle y, x\rangle) \in U$$

ovvero $HALT \leq U$.
$f$ é computabile. In particolare se $y$ = code($M$) costruiamo TM tale che essa o simula un'altra TM (rimanendo computabile) o entra in loop.

## Soluzione 3 (Continua)

Dimostriamo che $f$ é computabile e

$$\langle y, x\rangle \in HALT \quad \text{sse } f(\langle y, x\rangle) \in U$$

ovvero $HALT \leq U$.
$f$ é computabile. In particolare se $y = \text{code}(M)$ costruiamo TM tale che essa o simula un'altra TM (rimanendo computabile) o entra in loop. Caso

$y \neq \text{code}(M)$ triviale: se $y \neq \text{code}(M)$, allora $f(\langle y, x\rangle) \notin U$.

## Soluzione 3 (Continua)

Dimostriamo che $f$ é computabile e

$$\langle y,x\rangle \in HALT \quad \text{sse } f(\langle y,x\rangle) \in U$$

ovvero $HALT \leq U$.
$f$ é computabile. In particolare se $y = \text{code}(M)$ costruiamo TM tale che essa o simula un'altra TM (rimanendo computabile) o entra in loop. Caso

$y \neq \text{code}(M)$ triviale: se $y \neq \text{code}(M)$, allora $f(\langle y,x\rangle) \notin U$.

Caso $y = \text{code}(M)$:

## Soluzione 3 (Continua)

Dimostriamo che $f$ é computabile e

$$\langle y, x\rangle \in HALT \quad \text{sse } f(\langle y, x\rangle) \in U$$

ovvero $HALT \leq U$.
$f$ é computabile. In particolare se $y = \text{code}(M)$ costruiamo TM tale che essa o simula un'altra TM (rimanendo computabile) o entra in loop. Caso

$y \neq \text{code}(M)$ triviale: se $y \neq \text{code}(M)$, allora $f(\langle y, x\rangle) \notin U$.

Caso $y = \text{code}(M)$: $\langle y, x\rangle \in HALT$

## Soluzione 3 (Continua)

Dimostriamo che $f$ é computabile e

$$\langle y, x\rangle \in HALT \quad \text{sse } f(\langle y, x\rangle) \in U$$

ovvero $HALT \leq U$.
$f$ é computabile. In particolare se $y = \text{code}(M)$ costruiamo TM tale che essa o simula un'altra TM (rimanendo computabile) o entra in loop. Caso

$y \neq \text{code}(M)$ triviale: se $y \neq \text{code}(M)$, allora $f(\langle y, x\rangle) \notin U$.

Caso $y = \text{code}(M)$: $\langle y, x\rangle \in HALT$ sse $y = \text{code}(M)$ & ferma su $x$ sse

## Soluzione 3 (Continua)

Dimostriamo che $f$ é computabile e

$$\langle y, x\rangle \in HALT \quad \text{sse } f(\langle y, x\rangle) \in U$$

ovvero $HALT \leq U$.
$f$ é computabile. In particolare se $y = \text{code}(M)$ costruiamo TM tale che essa o simula un'altra TM (rimanendo computabile) o entra in loop. Caso

$y \neq \text{code}(M)$ triviale: se $y \neq \text{code}(M)$, allora $f(\langle y, x\rangle) \notin U$.

Caso $y = \text{code}(M)$: $\langle y, x\rangle \in HALT$ sse $y = \text{code}(M)$ & ferma su $x$ sse $M_{M,x}$ ferma su `111` (& $f(\langle y, x\rangle) = \text{code}(M_{M,x})$)

## Soluzione 3 (Continua)

Dimostriamo che $f$ é computabile e

$$\langle y, x\rangle \in HALT \quad \text{sse } f(\langle y, x\rangle) \in U$$

ovvero $HALT \leq U$.
$f$ é computabile. In particolare se $y = \text{code}(M)$ costruiamo TM tale che essa o simula un'altra TM (rimanendo computabile) o entra in loop. Caso

$y \neq \text{code}(M)$ triviale: se $y \neq \text{code}(M)$, allora $f(\langle y, x\rangle) \notin U$.

Caso $y = \text{code}(M)$: $\langle y, x\rangle \in HALT$ sse $y = \text{code}(M)$ & ferma su $x$ sse $M_{M,x}$ ferma su $\texttt{111}$ (& $f(\langle y, x\rangle) = \text{code}(M_{M,x})$) sse $f(\langle y, x\rangle) \in U$.

## Soluzione 3 (Continua)

Dimostriamo che $f$ é computabile e

$$\langle y, x\rangle \in HALT \quad \text{sse } f(\langle y, x\rangle) \in U$$

ovvero $HALT \leq U$.
$f$ é computabile. In particolare se $y = \text{code}(M)$ costruiamo TM tale che essa o simula un'altra TM (rimanendo computabile) o entra in loop. Caso

$y \neq \text{code}(M)$ triviale: se $y \neq \text{code}(M)$, allora $f(\langle y, x\rangle) \notin U$.

Caso $y = \text{code}(M)$: $\langle y, x\rangle \in HALT$ sse $y = \text{code}(M)$ & ferma su $x$ sse $M_{M,x}$ ferma su `111` (& $f(\langle y, x\rangle) = \text{code}(M_{M,x})$) sse $f(\langle y, x\rangle) \in U$.
Quindi $\langle y, x\rangle \in HALT$ sse $f(\langle y, x\rangle) \in U$.

# Linguaggi Triviali e Teorema di Rice

## Problema 4.

Quali delle seguenti sono proprietá di linguaggio triviali? Quali no?

a. $\{y \mid y = \text{code}(M)\ \&\ \epsilon \in L_M\}$
b. $\{y \mid y = \text{code}(M)\ \&\ M$ non ha stato iniziale$\}$
c. $\{y \mid y = \text{code}(M)\ \&\ L_M$ contiene tutte le stringhe di lunghezza pari$\}$
d. $\{y \mid y = \text{code}(M)\ \&\ M$ ha 3 stati$\}$.

## Correzione tra 10/15 minuti!

## Ricorda che...

**Definizione (Proprietá di Linguaggio)**

Una *proprietá di TM-linguaggio* é una funzione da insieme di TM a $\{0, 1\}$, tale che $L_M = L_{M'}$ implica $P(M) = P(M')$. Le TM che soddisfano $P$ sono indicate come

$$\{y \in \Sigma^* \mid y = \mathrm{code}(M) \ \& \ P(M) = 1\}.$$

**Definizione (Proprietá Non Triviale)**

Una proprietá di TM-linguaggio é *non triviale* se esiste una TM $M$ tale che $P(M) = 1$ e una TM $M'$ tale che $P(M') = 0$.

### Soluzione 4.

**Soluzione 4.**

a. É una proprietá non triviale.

**Soluzione 4.**

a. É una proprietá non triviale.
b. É una proprietá triviale.

## Soluzione 4.

a. É una proprietá non triviale.
b. É una proprietá triviale.
c. É una proprietá non triviale.

### Soluzione 4.

a. É una proprietá non triviale.
b. É una proprietá triviale.
c. É una proprietá non triviale.
d. Non é una proprietá di TM-linguaggio ma una proprietá della TM.

**Problema 5.**

Enuncia il teorema di Rice. Puoi applicarlo per dimostrare l'indecidibilitá di

$$INF = \{\text{code}(M) \mid M \text{ TM tale che } L(M) \text{ linguaggio infinito}\}?$$

**Correzione tra 15/20 minuti!**

**Ricorda che...**

### Teorema di Rice

Se $P$ é una proprietá non triviale, allora "il problema ha la proprietá $P$?" é indecidibile.

**Soluzione 5.**

Sí.

**Soluzione 5.**

Sí. *INF* soddisfa le condizioni richieste dal teorema di Rice:

## Soluzione 5.

Sí. *INF* soddisfa le condizioni richieste dal teorema di Rice:

1. É una proprietá del linguaggio: se due TM riconoscono lo stesso linguaggio o entrambe hanno descrizioni in INF o nessuna delle due ne ha.

**Soluzione 5.**

Sí. *INF* soddisfa le condizioni richieste dal teorema di Rice:

1. É una proprietá del linguaggio: se due TM riconoscono lo stesso linguaggio o entrambe hanno descrizioni in INF o nessuna delle due ne ha.
2. Non é triviale: alcune TM hanno linguaggi infiniti, altre no.

**Soluzione 5.**

Sí. *INF* soddisfa le condizioni richieste dal teorema di Rice:

1. É una proprietá del linguaggio: se due TM riconoscono lo stesso linguaggio o entrambe hanno descrizioni in INF o nessuna delle due ne ha.
2. Non é triviale: alcune TM hanno linguaggi infiniti, altre no.

Quindi, per il teorema di Rice, *INF* é indecidibile.