Esercitazioni I

Docente: Fabio Zanasi, Tutor: Melissa Antonelli

melissa.antonelli2@unibo.it

1 Marzo 2023

Prima di iniziare

- Esercitazioni: (alcuni) mercoledì 9:15-11:35 (pausa 10:30).
- Esercizi scaricabili da virtuale e presentati in slide.
- Svolgete gli esercizi autonomamente ma per ogni dubbio chiedete alla tutor.
- Alcune soluzioni verranno presentate in slide.
- Dal pomeriggio troverete le soluzioni su virtuale.
- Per ulteriori dubbi: `melissa.antonelli2@unibo.it`.

Sulla Codifica

Quesito 1.

Cosa si intende per codifica di un problema di decisione? Quali proprietá deve rispettare?

Problema 1.

Siano x una variabile, a, b, c interi e:

$$q(x) = ax^2 + bx + c.$$

Consideriamo il problema decisionale dato dall'insieme di quadratiche su interi avente risposta SI quando esiste un intero x tale che $q(x) = 0$.
Presentare un *sistema di codifica* per processare questo problema tramite TM.

Problema 1.

Siano x una variabile, a, b, c interi e:

$$q(x) = ax^2 + bx + c.$$

Consideriamo il problema decisionale dato dall'insieme di quadratiche su interi avente risposta SI quando esiste un intero x tale che $q(x) = 0$.
Presentare un *sistema di codifica* per processare questo problema tramite TM.

Suggerimento.

Ogni quadratica considerata é definita da tre interi a, b, e c, quindi é sufficiente un sistema di codifica per tre interi consecutivi.

Schema di Codifica

Abbiamo visto che...

Schema di Codifica

Abbiamo visto che...

* Per calcolare la risposta al *problema decisionale* serve un programma che riceva in input dati di tipo corretto e restituisca output SI/NO.

Schema di Codifica

Abbiamo visto che...

* Per calcolare la risposta al *problema decisionale* serve un programma che riceva in input dati di tipo corretto e restituisca output SI/NO.
* Le TM calcolano funzioni $\mathbb{N}^k \to \mathbb{N}$.

Schema di Codifica

Abbiamo visto che...

- Per calcolare la risposta al *problema decisionale* serve un programma che riceva in input dati di tipo corretto e restituisca output SI/NO.
- Le TM calcolano funzioni $\mathbb{N}^k \to \mathbb{N}$.
- Quindi, dobbiamo **codificare** il problema decisionale come funzione caratteristica di un linguaggio formale.

Schema di Codifica

Abbiamo visto che...

- Per calcolare la risposta al *problema decisionale* serve un programma che riceva in input dati di tipo corretto e restituisca output SI/NO.
- Le TM calcolano funzioni $\mathbb{N}^k \to \mathbb{N}$.
- Quindi, dobbiamo **codificare** il problema decisionale come funzione caratteristica di un linguaggio formale.

- Dato un alfabeto Σ, un *linguaggio formale* é un sottoinsieme di Σ^*.

Schema di Codifica

Abbiamo visto che...

- Per calcolare la risposta al *problema decisionale* serve un programma che riceva in input dati di tipo corretto e restituisca output SI/NO.
- Le TM calcolano funzioni $\mathbb{N}^k \to \mathbb{N}$.
- Quindi, dobbiamo **codificare** il problema decisionale come funzione caratteristica di un linguaggio formale.

- Dato un alfabeto Σ, un *linguaggio formale* é un sottoinsieme di Σ^*.
- La funzione caratteristica di L é $\chi_L : \Sigma^* \to \{0,1\}$ tale che:

$$\chi_L(x) = \begin{cases} 1 & \text{if } x \in L \\ 0 & \text{altrimenti.} \end{cases}$$

Schema di Codifica

Abbiamo visto che...

- Per calcolare la risposta al *problema decisionale* serve un programma che riceva in input dati di tipo corretto e restituisca output SI/NO.
- Le TM calcolano funzioni $\mathbb{N}^k \to \mathbb{N}$.
- Quindi, dobbiamo **codificare** il problema decisionale come funzione caratteristica di un linguaggio formale.

- Dato un alfabeto Σ, un *linguaggio formale* é un sottoinsieme di Σ^*.
- La funzione caratteristica di L é $\chi_L : \Sigma^* \to \{0, 1\}$ tale che:

$$\chi_L(x) = \begin{cases} 1 & \text{if } x \in L \\ 0 & \text{altrimenti.} \end{cases}$$

$$\textit{dato } \alpha \quad \overset{\text{schema di codifica}}{\Rightarrow} \quad \text{code}(\alpha) \in \Sigma^*$$

Schema di Codifica

Abbiamo visto che...

* Linguaggio che codifica problema di decisione é:

$$L = \{x \in \Sigma^* \mid x = \text{code}(\alpha)\text{per dato } \alpha \text{ e } \alpha \text{istanza positiva del problema}\}$$

* Proprietá di code:
  - se $\alpha \neq \beta$, allora code(α) $\neq$ code(β)
  - possiamo verificare se $x \in \Sigma^*$ é code(α), per qualche α
  - possiamo calcolare α a partire da code(α).

Soluzione 1.

Sistema di codifica con alfabeto $\Sigma = \{\mathtt{0}, \mathtt{1}\}$:

- ogni *n* (positivo) é codificato da stringa di `1` di lunghezza *n*
- ogni $-n$ (negativo) é codificato da uno `0` seguito da una stringa di `1` di lunghezza *n*
- 0 é codificato da `000`
- la tripla é codificata dalle codifiche di ciascuno dei suoi elementi separate da `0`.

Soluzione 1.

Sistema di codifica con alfabeto $\Sigma = \{0, 1\}$:

- ogni n (positivo) é codificato da stringa di 1 di lunghezza n
- ogni $-n$ (negativo) é codificato da uno 0 seguito da una stringa di 1 di lunghezza n
- 0 é codificato da 000
- la tripla é codificata dalle codifiche di ciascuno dei suoi elementi separate da 0.

Esempio.

La quadratica $x^2 - 2x + 3$ é codificata dalla striga:

Soluzione 1.

Sistema di codifica con alfabeto $\Sigma = \{0, 1\}$:

- ogni n (positivo) é codificato da stringa di 1 di lunghezza n
- ogni $-n$ (negativo) é codificato da uno 0 seguito da una stringa di 1 di lunghezza n
- 0 é codificato da 000
- la tripla é codificata dalle codifiche di ciascuno dei suoi elementi separate da 0.

Esempio.

La quadratica $x^2 - 2x + 3$ é codificata dalla striga:

$$100110111.$$

Macchine di Turing

Quesito 2.

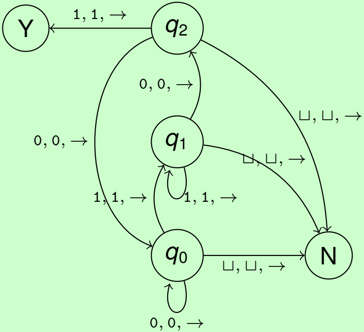
Definisci intuitivamente e formalmente la TM standard.

Problema 2.1.

Considera l'alfabeto $\Sigma = \{0, 1\}$ e definisci una TM tale che:

* se la stringa in input contiene 101, si ferma e accetta l'input
* altrimenti, rifiuta l'input.

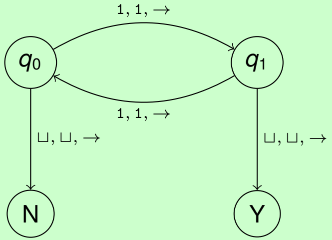
Soluzione 2.1.



Problema 2.2.

Rappresenta una TM che decide il linguaggio delle stringhe dispari su alfabeto $\Sigma = \{\mathtt{1}\}$.

Soluzione 2.2.



Problema 2.3.

Sia l'alfabeto Σ costituito da a, b, c.

a. Definisci una TM $\mathcal{T}$ tale che:
   - quando l'input contiene *ab*, si fermi e accetti l'input;
   - altrimenti, si fermi e rigetti l'input.

   (Sia l'input la stringa finita di caratteri in Σ seguita da $\sqcup$.)

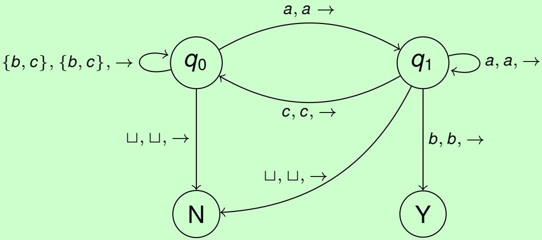
Problema 2.3.

Sia l'alfabeto Σ costituito da a, b, c.

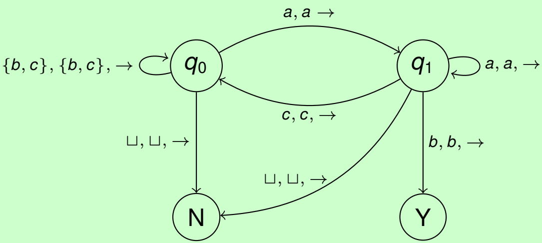
a. Definisci una TM $\mathcal{T}$ tale che:
   * quando l'input contiene *ab*, si fermi e accetti l'input;
   * altrimenti, si fermi e rigetti l'input.

   (Sia l'input la stringa finita di caratteri in Σ seguita da $\sqcup$.)
b. Quale *regular expression* descrive il linguaggio deciso da $\mathcal{T}$?

Soluzione 2.3.



Soluzione 2.3.



Le espressioni che descrivono il linguaggio deciso da $\mathcal{T}$ sono della forma:

$$(a|b|c)^* ab(a|b|c)^*.$$

Linguaggi Decidibili e Riconoscibili

Quesito 3.

Cosa si intende per linguaggio decidibile? E per linguaggio riconoscibile?

Problema 3.

Esiste un linguaggio che sia decidibile ma non riconoscibile?

Linguaggi Decidibili e Riconoscibili

Abbiamo visto che...

Linguaggi Decidibili e Riconoscibili

Abbiamo visto che...

* un linguaggio L si dice *decidibile* quando esiste una TM che decide L

Linguaggi Decidibili e Riconoscibili

Abbiamo visto che...

* un linguaggio L si dice *decidibile* quando esiste una TM che decide L
* un linguaggio L si dice *riconoscibile* quando esiste una TM che semi-decide L (cioé si ferma per ogni input $x \in L$ e non si ferma per ogni input $x \notin L$).

Soluzione 3.

No.

Soluzione 3.

No.

Assumiamo per assurdo tale L esista. Per decidibilitá deve esistere una TM $\mathcal{M}$ che decide L. In $\mathcal{M}$ sostituiamo lo stato N con loop infinito. Per definizione, questa nuova TM *riconosce* L, da cui si ha contraddizione.

Esistono invece linguaggi riconoscibili ma *non* decidibili.

Halting Problem

Quesito 4.

Enuncia la tesi di Church-Turing. Perché in questo contesto la TM é un modello computazionale concettualmente rilevante? Sapresti indicare almeno due modelli equivalenti alla TM standard?

Problema 4.

Dimostra per assurdo che $HALT^{-}$ non é riconoscibile da una TM.

Halting Problem

Abbiamo visto che...

- il linguaggio del problema della fermata

$$\mathrm{HALT} = \{\langle y, x\rangle \in \Sigma^* \times \Sigma^* \mid y = \mathrm{code}(\mathcal{M}) \;\&\; \mathcal{M} \text{ ferma su } x\}$$

  non é *decidibile*

- il complemento di HALT,

$$\begin{aligned}\mathrm{HALT}^{-} = \{\langle y, x\rangle \in \Sigma^* \times \Sigma^* \mid\ & y \neq \mathrm{code}(\mathcal{M}) \text{ per ogni } \mathcal{M} \text{ or} \\ & y = \mathrm{code}(\mathcal{M}) \;\&\; \mathcal{M} \text{ non ferma su } x\}.\end{aligned}$$

  non é *riconoscibile*.

Suggerimento.

La dimostrazione é per assurdo. Assumi $HALT^{-}$ sia riconoscibile quindi esista TM $\mathcal{M}_{H^{-}}$ che lo riconosce.

Suggerimento.

La dimostrazione é per assurdo. Assumi $HALT^-$ sia riconoscibile quindi esista TM $\mathcal{M}_{H^-}$ che lo riconosce. Definiamo nuova TM $\mathcal{M}''$ tale che, per ogni $z \in \Sigma^*$, esegue $\mathcal{M}_{H^-}$ su $\langle z, z \rangle$ e si ferma se la macchina si ferma; altrimenti entra in loop.

Soluzione 4.

Assumi $\mathcal{M}_{H^-}$ si fermi su $\langle$code($\mathcal{M}''$),code($\mathcal{M}''$)$\rangle$.

Soluzione 4.

Assumi $\mathcal{M}_{H-}$ si fermi su $\langle$code($\mathcal{M}''$),code($\mathcal{M}''$)$\rangle$. Per Df. di $\mathcal{M}''$, $\mathcal{M}''$ si ferma su code($\mathcal{M}''$).

Soluzione 4.

Assumi $\mathcal{M}_{H^-}$ si fermi su $\langle$code($\mathcal{M}''$),code($\mathcal{M}''$)$\rangle$. Per Df. di $\mathcal{M}''$, $\mathcal{M}''$ si ferma su code($\mathcal{M}''$). Per Df. di $HALT^-$, $\langle$code($\mathcal{M}''$), code($\mathcal{M}''$)$\rangle \notin HALT^-$.

Soluzione 4.

Assumi $\mathcal{M}_{H^-}$ si fermi su $\langle$code($\mathcal{M}''$),code($\mathcal{M}''$)$\rangle$. Per Df. di $\mathcal{M}''$, $\mathcal{M}''$ si ferma su code($\mathcal{M}''$). Per Df. di $HALT^-$, $\langle$code($\mathcal{M}''$), code($\mathcal{M}''$)$\rangle \notin HALT^-$. Per Df. di $\mathcal{M}_{H^-}$, $\mathcal{M}_{H^-}$ non si ferma su $\langle$ code($\mathcal{M}''$), code($\mathcal{M}''$)$\rangle$.

Soluzione 4.

Assumi $\mathcal{M}_{H^-}$ si fermi su $\langle$code($\mathcal{M}''$),code($\mathcal{M}''$)$\rangle$. Per Df. di $\mathcal{M}''$, $\mathcal{M}''$ si ferma su code($\mathcal{M}''$). Per Df. di $HALT^-$, $\langle$code($\mathcal{M}''$), code($\mathcal{M}''$)$\rangle \notin HALT^-$. Per Df. di $\mathcal{M}_{H^-}$, $\mathcal{M}_{H^-}$ non si ferma su $\langle$ code($\mathcal{M}''$), code($\mathcal{M}''$)$\rangle$.
$\Rightarrow$ Contraddizione.

Soluzione 4. (continua)

Assumi $\mathcal{M}_{H^-}$ non si fermi su $\langle$ code($\mathcal{M}''$), code$\mathcal{M}''\rangle$.

Soluzione 4. (continua)

Assumi $\mathcal{M}_{H-}$ non si fermi su $\langle$ code($\mathcal{M}''$), code$\mathcal{M}''\rangle$. Per Df. di $\mathcal{M}''$, $\mathcal{M}''$ non si ferma su code($\mathcal{M}''$).

Soluzione 4. (continua)

Assumi $\mathcal{M}_{H^-}$ non si fermi su $\langle$ code($\mathcal{M}''$), code$\mathcal{M}''\rangle$. Per Df. di $\mathcal{M}''$, $\mathcal{M}''$ non si ferma su code($\mathcal{M}''$). Per Df. di $HALT^-$, $\langle$code($\mathcal{M}''$), code($\mathcal{M}''$)$\rangle \in HALT^-$.

Soluzione 4. (continua)

Assumi $\mathcal{M}_{H^-}$ non si fermi su $\langle$ code($\mathcal{M}''$), code$\mathcal{M}''\rangle$. Per Df. di $\mathcal{M}''$, $\mathcal{M}''$ non si ferma su code($\mathcal{M}''$). Per Df. di *HALT*$^-$, $\langle$code($\mathcal{M}''$), code($\mathcal{M}''$)$\rangle \in$ *HALT*$^-$. Per Df. di $\mathcal{M}_{H^-}$ concludiamo che $\mathcal{M}_{H^-}$ si ferma su $\langle$code($\mathcal{M}''$), code($\mathcal{M}''$)$\rangle$.

Soluzione 4. (continua)

Assumi $\mathcal{M}_{H^-}$ non si fermi su $\langle$ code($\mathcal{M}''$), code$\mathcal{M}''\rangle$. Per Df. di $\mathcal{M}''$, $\mathcal{M}''$ non si ferma su code($\mathcal{M}''$). Per Df. di $HALT^-$, $\langle$code($\mathcal{M}''$), code($\mathcal{M}''$)$\rangle \in HALT^-$. Per Df. di $\mathcal{M}_{H^-}$ concludiamo che $\mathcal{M}_{H^-}$ si ferma su $\langle$code($\mathcal{M}''$), code($\mathcal{M}''$)$\rangle$.
$\Rightarrow$ Contraddizione.

Soluzione 4. (continua)

Assumi $\mathcal{M}_{H^-}$ non si fermi su $\langle$ code($\mathcal{M}''$), code$\mathcal{M}''\rangle$. Per Df. di $\mathcal{M}''$, $\mathcal{M}''$ non si ferma su code($\mathcal{M}''$). Per Df. di $HALT^-$, $\langle$code($\mathcal{M}''$), code($\mathcal{M}''$)$\rangle \in HALT^-$. Per Df. di $\mathcal{M}_{H^-}$ concludiamo che $\mathcal{M}_{H^-}$ si ferma su $\langle$code($\mathcal{M}''$), code($\mathcal{M}''$)$\rangle$.
$\Rightarrow$ Contraddizione.

L'unica assunzione per costruire $\mathcal{M}''$é l'esistenza di una TM $\mathcal{M}_{H^-}$ che riconosce $HALT^-$.

Soluzione 4. (continua)

Assumi $\mathcal{M}_{H-}$ non si fermi su $\langle$ code($\mathcal{M}''$), code$\mathcal{M}''\rangle$. Per Df. di $\mathcal{M}''$, $\mathcal{M}''$ non si ferma su code($\mathcal{M}''$). Per Df. di $HALT^-$, $\langle$code($\mathcal{M}''$), code($\mathcal{M}''$)$\rangle \in HALT^-$. Per Df. di $\mathcal{M}_{H-}$ concludiamo che $\mathcal{M}_{H-}$ si ferma su $\langle$code($\mathcal{M}''$), code($\mathcal{M}''$)$\rangle$.
$\Rightarrow$ Contraddizione.

L'unica assunzione per costruire $\mathcal{M}''$é l'esistenza di una TM $\mathcal{M}_{H-}$ che riconosce $HALT^-$.

Poiché $\mathcal{M}_{H-}$ non esiste, $HALT^-$ non é riconoscibile.