2° PARTE

- MACCHINE DI TURING $\boxed{\text{Dimostrazione}}$ → FORMALE
- INDECIDIBILITÀ Dimostra L indecidibile
- DECIDIBILITÀ Dimostra L decidibile
- NP-HARD Dimostrazione

MACCHINA DI TURING

→ COME SI MUOVE (FUNZIONE DI TRANSIZIONE)

VARIANTE (L, R, ALTRE COSE) ← (→) TM SINGOLO (L, R)

$\{\uparrow, \downarrow, \leftarrow, \rightarrow\}$

1. Una *macchina di Turing con reset a sinistra* è una variante delle comuni macchine di Turing, dove la funzione di transizione ha la forma:

$$\delta : Q \times \Gamma \mapsto Q \times \Gamma \times \{R, RESET\}.$$

→ $\{L, R\}$

Se $\delta(q, a) = (r, b, RESET)$, quando la macchina si trova nello stato q e legge a, la testina scrive b sul nastro, salta all'estremità sinistra del nastro ed entra nello stato r. Per sapere su quale cella saltare la macchina usa il simbolo speciale $\triangleright$ per identificare l'estremità di sinistra del nastro. Questo simbolo si può trovare solo in una cella del nastro, e non può essere sovrascritto o cancellato. La computazione di una macchina di Turing con reset a sinistra sulla parola w inizia con $\triangleright w$ sul nastro. Si noti che queste macchine non hanno la solità capacità di muovere la testina di una cella a sinistra.

Mostrare che le macchine di Turing con reset a sinistra riconoscono la classe dei linguaggi Turing-riconoscibili.

δ → "Q" = INPUT ("b" = OUTPUT

(DELTA = FUNZIONE DI TRANSIZIONE) = COME SI MUOVE

TURING - RICONOSCIBILE

DESCRIVE IL PROBLEMA = [TM CHE ESPRIME L COME LINGUAGGIO]

DECIDIBILE → T. FINITO (PD) DIMOSTRA IL PROBLEMA

INDECIDIBILE → PROBLEMA CHE È IMPOSSIBILE

$\{L, R, D\}$

$\{L, R\}$

TM SINGOLO

(a)

TM RESET A SX

1. Una *macchina di Turing con reset a sinistra* è una variante delle comuni macchine di Turing, dove la funzione di transizione ha la forma:

$$\delta : Q \times \Gamma \mapsto Q \times \Gamma \times \{R, RESET\}. \rightarrow \{L, R\}$$

Se $\delta(q, a) = (r, b, RESET)$, quando la macchina si trova nello stato q e legge a, la testina scrive b sul nastro, salta all'estremità sinistra del nastro ed entra nello stato r. Per sapere su quale cella saltare la macchina usa il simbolo speciale $\triangleright$ per identificare l'estremità di sinistra del nastro. Questo simbolo si può trovare solo in una cella del nastro, e non può essere sovrascritto o cancellato. La computazione di una macchina di Turing con reset a sinistra sulla parola w inizia con $\triangleright w$ sul nastro. Si noti che queste macchine non hanno la solità capacità di muovere la testina di una cella a sinistra.

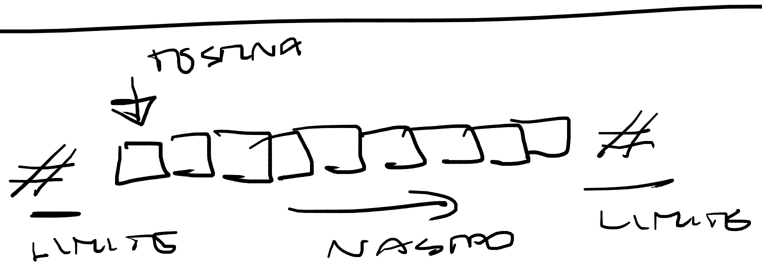
Mostrare che le macchine di Turing con reset a sinistra riconoscono la classe dei linguaggi Turing-riconoscibili.

→ ANDATA(?)

INSIEME DI TUTTI GLI INPUT

TUTTE LE TM RISOLVONO IL PROBLEMA

TM A NASTRO SINGOLO



[TM A RESET A SX]

$\delta\{L, R\} = \delta\{R, RESET\}$

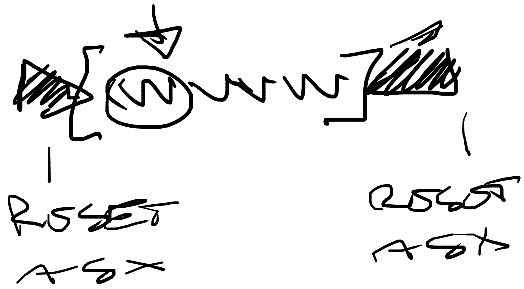
DA DIMOSTRARE

- $\delta(q, a) = (r, b, R)$

LA MACCHINA SI MUOVE A DX E VA NUOVO STATO R SCRIVENDO "b"

- $\delta(q, a) = (r, b, RESET)$

W = STRINGA



DISTINGUIAMO SE RESET A SX O DX METTENDO

- RAGGIUNGO RESET
- METTO BLANK SULLA CELLA
- CONTINUO NELLA TRANSIZIONE

⊔ = BLANK

M = MDT SINGOLO

(a) Mostriamo come convertire una TM con reset a sinistra M in una TM standard S equivalente. S simula il comportamento di M nel modo seguente. Se la mossa da simulare prevede uno spostamento a destra, allora S esegue direttamente la mossa. Se la mossa prevede un $RESET$, allora S scrive il nuovo simbolo sul nastro, poi scorre il nastro a sinistra finché non trova il simbolo $\triangleright$, e riprende la simulazione dall'inizio del nastro. Per ogni stato q di M, S possiede uno stato q_{RESET} che serve per simulare il reset e riprendere la simulazione dallo stato corretto.

→ S = "Su input $\boxed{w}$: ← STRINGA GENERICA → ABBAA $\Sigma = \{A, B\}^*$ A

1. scrive il simbolo $\triangleright$ subito prima dell'input, in modo che il nastro contenga $\triangleright w$.
2. Se la mossa da simulare è $\delta(q, a) = (r, b, R)$, allora S la esegue direttamente: scrive b sul nastro, muove la testina a destra e va nello stato r.
3. Se la mossa da simulare è $\boxed{\delta(q, a) = (r, b, RESET),}$ allora S esegue le seguenti operazioni: scrive b sul nastro, poi muove la testina a sinistra e va nello stato r_{RESET}. La macchina rimane nello stato r_{RESET} e continua a muovere la testina a sinistra finché non trova il simbolo $\triangleright$. A quel punto la macchina sposta la testina un'ultima volta a sinistra, poi di una cella a destra per tornare sopra al simbolo di fine nastro. La computazione riprende dallo stato r.
4. Se non sei nello stato di accettazione o di rifiuto, ripeti da 2."

(a) SINGOLO → RESET A SX

{L, R} {R, RESET}

W = ABBAA → PAROLA

ABBAA

SINGOLO ← RESET A SX

(b)

{L, R}

RESET

▷ www ◁

(b) Mostriamo come convertire una TM standard S in una TM con reset a sinistra M equivalente. M simula il comportamento di S nel modo seguente. Se la mossa da simulare prevede uno spostamento a destra, allora M può eseguire direttamente la mossa. Se la mossa da simulare prevede uno spostamento a sinistra, allora M simula la mossa come descritto dall'algoritmo seguente. L'algoritmo usa un nuovo simbolo $\triangleleft$ per identificare la fine della porzione di nastro usata fino a quel momento, e può marcare le celle del nastro ponendo un punto al di sopra di un simbolo.

M = "Su input w:

1. Scrive il simbolo $\triangleleft$ subito dopo l'input, per marcare la fine della porzione di nastro utilizzata. Il nastro contiene $\triangleright w \triangleleft$.
2. Simula il comportamento di S. Se la mossa da simulare è $\delta(q, a) = (r, b, R)$, allora M la esegue direttamente: scrive b sul nastro, muove la testina a destra e va nello stato r. Se muovendosi a destra la macchina si sposta sulla cella che contiene $\triangleleft$, allora questo significa che S ha spostato la testina sulla parte di nastro vuota non usata in precedenza. Quindi M scrive un simbolo blank marcato su questa cella, sposta $\triangleleft$ di una cella a destra, e fa un reset a sinistra. Dopo il reset si muove a destra fino al blank marcato, e prosegue con la simulazione mossa successiva. $\triangleright \dot{a} \triangleleft$
3. Se la mossa da simulare è $\delta(q, a) = (r, b, L)$, allora S esegue le seguenti operazioni:
   3.1 scrive b sul nastro, marcandolo con un punto, poi fa un reset a sinistra
   3.2 Se il simbolo subito dopo $\triangleright$ è già marcato, allora vuol dire che S ha spostato la testina sulla parte vuota di sinistra del nastro. Quindi M scrive un blank e sposta il contenuto del nastro di una cella a destra finché non trova il simbolo di fine nastro $\triangleleft$. Fa un reset a sinistra e prosegue con la simulazione della prossima mossa dal nuovo blank posto subito dopo l'inizio del nastro. Se il simbolo subito dopo $\triangleright$ non è marcato, lo marca, resetta a sinistra e prosegue con i passi successivi.

3.3 Si muove a destra fino al primo simbolo marcato, e poi a destra di nuovo.

3.4 se la cella in cui si trova è marcata, allora è la cella da cui è partita la simulazione. Toglie la marcatura e resetta. Si muove a destra finché non trova una cella marcata. Questa cella è quella immediatamente precedente la cella di partenza, e la simulazione della mossa è terminata

3.5 se la cella in cui si trova non è marcata, la marca, resetta, si muove a destra finché non trova una marcatura, cancella la marcatura e riprende da 3.3.

4. Se non sei nello stato di accettazione o di rifiuto, ripeti da 2."

ESERCIZIO

1. **(12 punti)** Una *Macchina di Turing con inserimento* è una macchina di Turing deterministica a nastro singolo che può inserire nuove celle nel nastro. Formalmente la funzione di transizione è definita come

$$\delta : Q \times \Gamma \mapsto Q \times \Gamma \times \{L, R, I\}$$

dove L, R indicano gli spostamenti a sinistra e a destra della testina, e I indica l'inserimento di una nuova cella nella posizione corrente della testina. Dopo una operazione di inserimento, la cella inserita contiene il simbolo blank, mentre la cella che si trovava sotto la testina si trova immediatamente a destra della nuova cella.

Dimostra che qualsiasi macchina di Turing con inserimento può essere simulata da una macchina di Turing deterministica a nastro singolo.

2

$\{L, R\} \rightarrow \{L, R, I\}$

SINGOLO $\rightarrow$ INSERIMENTO

Soluzione. Mostriamo come convertire una macchina di Turing con Inserimento M in una TM deterministica a nastro singolo S equivalente.

$\Sigma = \{a, b\}^* \rightarrow w = \{abbba\}$

$S =$ "Su input w:

1. Inizialmente S mette il suo nastro in un formato che gli consente di implementare l'operazione di inserimento di una cella, segnando con il simbolo speciale # la fine della porzione di nastro usata dalla macchina. Se w è l'input della TM, la configurazione iniziale del nastro è $w\#$.
2. La simulazione delle mosse del tipo $\delta(q, a) = (r, b, L)$ procede come nella TM standard: S scrive b sul nastro e muove la testina di una cella a sinistra.
3. La simulazione delle mosse del tipo $\delta(q, a) = (r, b, R)$ procede come nella TM standard: S scrive b sul nastro e muove la testina di una cella a destra. Se lo spostamento a destra porta la testina sopra il # che marca la fine del nastro, S scrive un blank $\sqcup$ al posto del #, e scrive un # nella cella immediatamente più a destra. La simulazione continua con la testina in corrispondenza del blank.
4. Per simulare una mossa del tipo $\delta(q, a) = (r, b, I)$ la TM S scrive un blank marcato nella cella corrente e sposta il contenuto del nastro, dalla cella corrente fino al # di fine nastro, di una cella più a destra. Quindi riporta la testina in corrispondenza del blank marcato, toglie la marcatura e scrive b nella cella immediatamente più a destra. La simulazione continua con la testina in corrispondenza della cella inserita.
5. Se in qualsiasi momento la simulazione raggiunge lo stato di accettazione di M, allora S termina con accettazione. Se in qualsiasi momento la simulazione raggiunge lo stato di rifiuto di M, allora S termina con rifiuto. Negli altri casi continua la simulazione dal punto 2."

1. **(12 punti)** Una *Macchina di Turing con inserimento* è una macchina di Turing deterministica a nastro singolo che può inserire nuove celle nel nastro. Formalmente la funzione di transizione è definita come

$$\delta : Q \times \Gamma \mapsto Q \times \Gamma \times \{L, R, I\}$$

VARIANTE...

dove L, R indicano gli spostamenti a sinistra e a destra della testina, e I indica l'inserimento di una nuova cella nella posizione corrente della testina. Dopo una operazione di inserimento, la cella inserita contiene il simbolo blank, mentre la cella che si trovava sotto la testina si trova immediatamente a destra della nuova cella.

Dimostra che qualsiasi macchina di Turing con inserimento può essere simulata da una macchina di Turing deterministica a nastro singolo.

SINGOLO → INSERIMENTO (⇐)

$\{L, R\}$ $\{L, R, I\}$

SINGOLO ⇐ INSERIMENTO

LO CONTIENE

$\{L, R, I\} \subseteq \{L, R\}$ OVVIO

IMPOSSIBILE → NON LO RISOLVE...

Teorema: A_{TM} è indecidibile

$$A_{TM} = \{\langle M, w\rangle \mid M \text{ è una TM che accetta la stringa } w\}$$

Dimostrazione:

- per contraddizione. Assumiamo A_{TM} decidibile per poi trovare una contraddizione
- Supponiamo H decisore per A_{TM}
- Cosa fa H con input $\langle M, w\rangle$?

$$H(\langle M, w\rangle) = \begin{cases} \text{accetta} & \text{se } M \text{ accetta } w \\ \text{rifiuta} & \text{se } M \text{ non accetta } w \end{cases}$$

Teorema: A_{TM} è indecidibile

- Definiamo una TM D che usa H come subroutine
- D = "Su input $\langle M\rangle$, dove M è una TM:
  1. Esegue H su input $\langle M, \langle M\rangle\rangle$
  2. Dà in output l'opposto dell'output di H. Se H accetta, rifiuta; se H rifiuta, accetta."
- Cosa fa D con input $\langle D\rangle$?

$$D(\langle D\rangle) = \begin{cases} \text{accetta} & \text{se } D \text{ non accetta } \langle D\rangle \\ \text{rifiuta} & \text{se } D \text{ accetta } \langle D\rangle \end{cases}$$

- Contraddizione!

1. H accetta $\langle M, w\rangle$ esattamente quando M accetta w
   a. Banale: abbiamo assunto che H esista e decida A_{TM}
   b. M rappresenta qualsiasi TM e w è una qualsiasi stringa
2. D rifiuta $\langle M\rangle$ esattamente quando M accetta $\langle M\rangle$
   a. Cosa è successo a w?
   b. w è solo una stringa, come $\langle M\rangle$. Tutto ciò che stiamo facendo è definire quale stringa dare in input alla macchina.
3. D rifiuta $\langle D\rangle$ esattamente quando D accetta $\langle D\rangle$
   a. Questa è la contraddizione.

NON SAPPIAMO SEMPRE GLI INPUT

↓

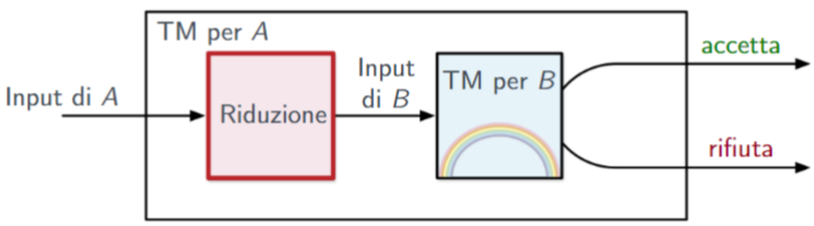
NON SAPPIAMO SE IL PROBLEMA SI RISOLVE SEMPRE

Il problema della fermata

$$HALT_{TM} = \{\langle M, w\rangle \mid M \text{ è una TM che si ferma su input } w\}$$

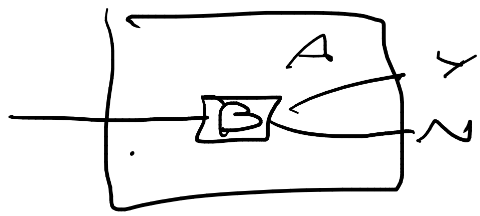
Riducibilità mediante funzione

Se esiste una riduzione da A a B, possiamo risolvere A usando una soluzione per B:



$A \le_m B$

MAPPING



SUPER $\le_m$ SUB

A_{TM} w $\le_m$ PROBLEMA

$HALT_{TM} \le_m$

A È INDECIDIBILE

↓

B È INDECIDIBILE

INDECIDIBILI

↓

Il problema del vuoto

$$E_{TM} = \{\langle M \rangle \mid M \text{ è una TM tale che } L(M) = \emptyset\}$$

Il problema della fermata

$$HALT_{TM} = \{\langle M, w \rangle \mid M \text{ è una TM che si ferma su input } w\}$$

$$A_{TM} = \{\langle M, w \rangle \mid M \text{ è una TM che accetta la stringa } w\}$$

STRINGA/PAROLA

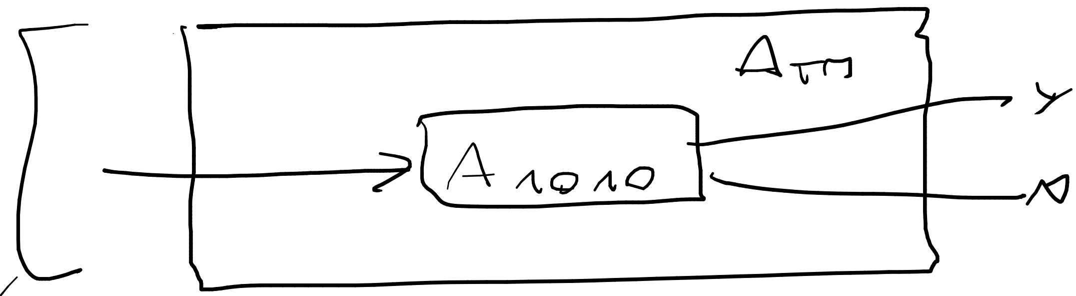
↓

COMBO DI CARATTERI

$w \to 1010$

2. *Dimostra che il seguente linguaggio è indecidibile:*

$$A_{1010} = \{\langle M \rangle \mid M \text{ è una TM tale che } \underline{1010} \in L(M)\}.$$



RIDUZIONE

$\underline{A_{TM}} \le_m \underline{A_{1010}}$

INDECIDIBILE

LO DIVENTA

2. *Dimostra che il seguente linguaggio è indecidibile:*

$$A_{1010} = \{\langle M \rangle \mid M \text{ è una TM tale che } 1010 \in L(M)\}.$$

Dimostriamo che A_{1010} è un linguaggio indecidibile mostrando che A_{TM} è riducibile ad A_{1010}. La funzione di riduzione f è calcolata dalla seguente macchina di Turing:

$w \in A \iff f(w) \in B$

$A \le_m B$

$A_{TM} \le_m A_{1010}$

$F =$ "su input $\langle M, w \rangle$, dove M è una TM e w una stringa:

1. Costruisci la seguente macchina M_w:
   $M_w =$ "su input x:
   1. Se $x \neq 1010$, rifiuta.
   2. Se $x = 1010$, esegue M su input w.
   3. Se M accetta, *accetta*.
   4. Se M rifiuta, *rifiuta*."
2. Restituisci $\langle M_w \rangle$."

A_{TM}

A_{1010}

w

$1010 =$

A

R

F = FUNZIONE DI RIDUZIONE

Dimostriamo che f è una funzione di riduzione da A_{TM} ad A_{1010}.

- Se $\langle M, w \rangle \in A_{TM}$ allora la TM M accetta w. Di conseguenza la macchina M_w costruita dalla funzione accetta la parola 1010. Quindi $f(\langle M, w \rangle) = \langle M_w \rangle \in A_{1010}$.
- Viceversa, se $\langle M, w \rangle \notin A_{TM}$ allora la computazione di M su w non termina o termina con rifiuto. Di conseguenza la macchina M_w rifiuta 1010 e $f(\langle M, w \rangle) = \langle M_w \rangle \notin A_{1010}$.

$w \in A \iff f(w) \in B$

$\langle M, w \rangle \in A_{TM} \iff \langle M_w \rangle \in A_{1010}$

2. (12 punti) Data una Turing Machine M, definiamo

$$\text{HALTS}(M) = \{w \mid M \text{ termina la computazione su } w\}.$$

LINGUAGGIO

Considera il linguaggio

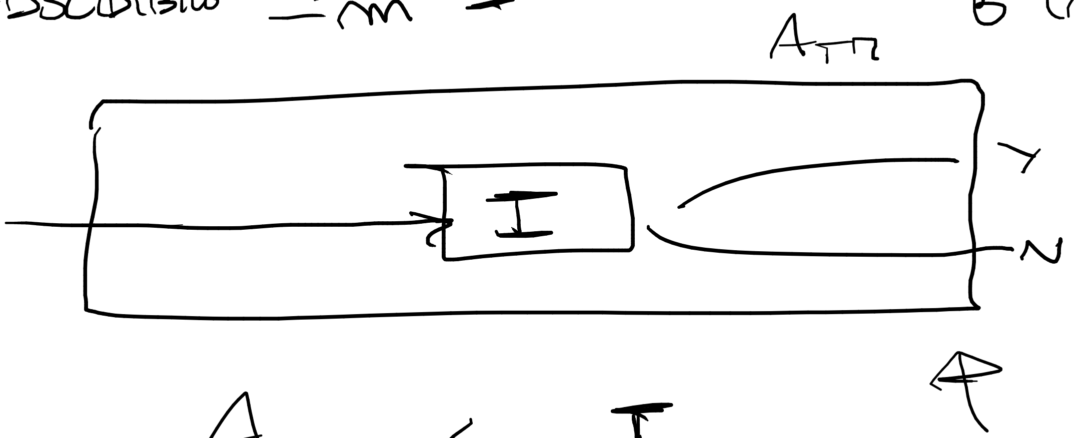
$$I = \{\langle M \rangle \mid \text{HALTS}(M) \text{ è un insieme infinito}\}.$$

Dimostra che I è indecidibile.

M SI FERMA MA IL LINGUAGGIO È INFINITO

A_{TM} E_{TM} $HALT_{TM}$

INDECIDIBILE $\leq_m$ I



$A_{TM} \leq_m I$

F → $\langle M, w \rangle$ = RIDUZIONE

TM STRINGA

Soluzione. La seguente macchina F calcola una riduzione mediante funzione $A_{TM} \leq_m I$:

M'

$w \in A_{TM} \iff f(\langle w \rangle) \in I$

RICONOSCE w

F = "su input $\langle M, w \rangle$, dove M è una TM e w una stringa:

1. Costruisci la seguente macchina M':

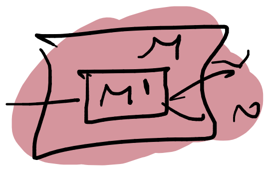
   M' = "Su input x:
   1. Esegue M su input w.
   2. Se M accetta, *accetta*.
   3. Se M rifiuta, va in loop."

2. Ritorna $\langle M' \rangle$."

Mostriamo che F calcola una funzione di riduzione f da A_{TM} a I, cioè una funzione tale che

$$\langle M, w \rangle \in A_{TM} \text{ se e solo se } M' \in I.$$

∅ INDECIDIBILE



- Se $\langle M, w \rangle \in A_{TM}$ allora la macchina M accetta w. In questo caso la macchina M' accetta tutte le parole, quindi $\text{HALTS}(M) = \Sigma^*$ che è un insieme infinito. Di conseguenza $M' \in I$.
- Viceversa, se $\langle M, w \rangle \notin A_{TM}$, allora la macchina M su input w rifiuta oppure va in loop. In entrambi i casi la macchina M' va in loop su tutte le stringhe, quindi $\text{HALTS}(M) = \emptyset$ che è un insieme finito. Di conseguenza $M' \notin I$.