Macchina di Turing

1)

$$\delta: Q \ x \ \Gamma \mapsto \mathrm{Q} \ \mathrm{x} \ \Gamma \ \mathrm{x} \ \{\mathrm{L, R, J}\}$$

2)

TM (Macchina di Turing) a nastro singolo → TM spreca-nastro

Da TM a nastro singolo ci spostiamo verso quelle sprecanastro con le stesse azioni, ma quelle sprecanastro hanno il salto J. Le macchine normali non saltano mai oltre la parte vuota

TM (Macchina di Turing) sprecanastro → TM a nastro singolo

Input w:

- La macchina deve marcare l'inizio dell'input w con un simbolo, per esempio con #w#

1) $\delta(q, a) = (r, b, L)$

La macchina S scrive b sul nastro e sposta la testina di una cella a sinistra: Se incontra la fine (un simbolo #) allora lascia inalterato il nastro

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

La macchina S scrive b sul nastro e sposta la testina di una cella a sinistra: Se incontra la fine (un simbolo #), usa un simbolo ∘ (blank) per dire "sono alla fine il nastro"

3) $\delta(q, a) = (r, b, J)$

Andiamo avanti finché possibile (verso #)

Torno indietro finché non trovo un blank (∘)

Ha capito che ha percorso tutta la macchina; essendo la cella non vuota, la "marca", realizza il salto "del doppio";

---- # ∘

Continua così finché non ho finito tutte le stringhe: se raggiungo la fine accetto, altrimenti rifiuto

4)

$$A \leq_m B$$

- Decidibile
  - Esiste una macchina con un linguaggio decidibile
- Indecidibile
  - Esiste una macchina con un linguaggio indecidibile

3)

a)

$SO_{TM} = \{< M >|\ M$ è una TM con alfabeto $\Sigma = \{1,2, \dots .9\}$ che accetta parole ordinate$\}$

Descrivere con un linguaggio vuol dire usare una macchina di Turing che accetta il linguaggio.

b)

$$A \leq_m B$$

Se B ha delle proprietà, valgono per A

Chiamiamo il linguaggio

$$A_{TM}$$

Questo problema indecidibile dice:

- hai un input
- se accetta, accetta
- se rifiuta, rifiuta

$$\overline{A_{TM}}$$

Questo problema indecidibile dice:

- hai un input
- se accetta, rifiuta
- se rifiuta, accetta

Riduzione $\overline{A_{TM}} \leq_M A$

F = su input $< M, w >$, dove M è una TM e w è una stringa:

- Costruisco una macchina di Turing
  - M' su input x
    - Se $x = 7865$ accetta
    - Se l'input è $x = 12345$ rifiutqa
  - In tutti gli altri casi rifiuta
- Ritorna M'

$< M, w > \in \overline{A_{TM}}$ se e solo se $< M' > \in SO_{TM}$

- Se $\in \overline{A_{TM}}$ accetta e in questo caso l'input accetta una parola disordinata
- Se $\notin \overline{A_{TM}}$ il linguaggio è ordinato allora la macchina rifiuta

Essendo che siamo partiti da un problema indecidibile, il linguaggio è indecidibile

$$E_{TM}$$

Questo problema decidibile dice:

- hai un input
- se l'input è vuoto, accetta
- se c'è qualche input, rifiuta

- Grammatica context-free

Più casi:

- Parto da un linguaggio e dimostro che è context-free

Uso una grammatica con simboli iniziali, finali e terminali

- Parto da qualcosa che è context-free

Esiste un DFA in grado di esprimere questo/una Macchina di Turing

a) $PERSISTENT_{CFG} = \{< G, A >|$ G è una grammatica context-free, mentre A è una variabile persistente

b)

Uso una macchina di Turing (decidibile = usare qualcosa di decidibile che esprime il linguaggio)

$N = TM$ che decide questo problema

Per fare in modo che sia decidibile uso un problema decidibile

E_{CFG} = se non c'è nessuna stringa è decidibile (accetta), altrimenti no

$N =$ su input $< G, A >$ dove G è la CFG, A è una variabile

- Verifica che A appartenga a G (rispetti le proprietà del linguaggio)
- Costruisco una CFG G' che elimina tutte le regole di G
- Eseguo la mia macchina N sull'input

Se abbiamo fatto questo passaggio:

- Se abbiamo l'output vuol dire che abbiamo tolto tutte le stringhe e va bene
- Se abbiamo l'input, questo risulta decidibile se uso G' per togliere tutte le regole e arrivo a uno stato finale

Pumping Lemma

$$w = xyz$$

$$y \neq \epsilon, |xy| \leq k$$

$$uvvu \in \{0,1\}^*$$

$$w = 0^k 110^k$$

Siccome $|xy| \leq k$,

$$x = 0^q$$

$$y = 0^p$$

$$z = 0^{k-p-q} 110^k$$

$$xy^2z = 0^q 0^{2p} 0^{k-p-q} = 0^{k+p} 110^k$$

Essendo il numero di 0 iniziali sbilanciati rispetto a quelli finali, allora il linguaggio non è regolare.

Esempio linguaggio non regolare

$$L = \{0^n 1^n\}$$

$$xyz = 0^p 1^p$$

$$xyz = 0^k 1^k 0^{p.-k}$$

$$x = 0^k$$

$$y = 1^k$$

$$z = 0^{p-k}$$

$$xy^2 z = 0^k 1^k 0^{2.-k}$$

Avrai un numero di 0 diverso rispetto agli 1 (ma anche rispetto agli 0 iniziali)

1)

Il linguaggio non deve avere prefissi:

- usiamo un DFA A' che ogni volta che prova a mettermi il prefisso, uso uno stato "isolato" terminante

$$A = (Q', \Sigma, \delta', {q_0}', F')$$

Aggiungiamo uno stato q_s

- $Q' = Q \cup \{q_s\}$
- Stesso alfabeto Σ
- $\delta'(q,a) = \begin{cases} \delta(q,a), & se\ q \notin F \\ q_s, & altrimenti \end{cases}$
- $q_0' = q_0$ (stesso stato iniziale)
- $F' = F$ (gli stati finali sono gli stessi)

Abbiamo due casi:

- se $w \in NOPREFIX(L)$, nel linguaggio non ci sono prefissi e tutti i prefissi portano a degli stati non accettanti abbiamo usato qualcosa che ci ha tolto i prefissi (quindi l'automa)
- se w è accettata dal DFA A' vuol dire che non ci sono prefissi e abbiamo una computazione accettante (rifiutiamo tutti i prefissi)