2. (12 punti) I *grawlix* sono sequenze di simboli senza senso che sostituiscono le parolacce nei fumetti.

#*$@ → LING. VOLGARE

Un linguaggio è *volgare* se contiene almeno un grawlix. Considera il problema di determinare se il linguaggio di una TM è volgare.

(a) Formula questo problema come un linguaggio $GROSS_{TM}$.
(b) Dimostra che il linguaggio $GROSS_{TM}$ è indecidibile.

$$\Sigma = \{\#, *, \$, @, !, €\}$$

SIA UNA PAROLA

SIA UN INSIEME DI PAROLE

$$[\, w = \# = \text{GRAWLIX} \,]$$

SCALETTA INDECIDIBILE:

(1) IMPOSTO PROBLEMA E RIDUZIONE

$$\left.\begin{array}{l} A_{TM} \\ E_{TM} \\ EQ_{TM} \end{array}\right] \leq_m \text{ESERCIZIO } GROSS_{TM}$$

COMPLEMENTO

$$\overline{A_{TM}} \,/\, \overline{E_{TM}} \,/\, \overline{EQ_{TM}}$$

2. (12 punti) I *grawlix* sono sequenze di simboli senza senso che sostituiscono le parolacce nei fumetti.

Un linguaggio è *volgare* se contiene almeno un grawlix. Considera il problema di determinare se il linguaggio di una TM è volgare.

(a) Formula questo problema come un linguaggio $GROSS_{TM}$.
(b) Dimostra che il linguaggio $GROSS_{TM}$ è indecidibile.

$GROSS_{TM} = \{ \langle M, w \rangle \mid M$ è una MdT e w contiene almeno un GRAWLIX $\}$

→ $L(w)$ è VOLGARE

(1) DEFINIZIONE PROBLEMA

2. (12 punti) I *grawlix* sono sequenze di simboli senza senso che sostituiscono le parolacce nei fumetti.

Un linguaggio è *volgare* se contiene almeno un grawlix. Considera il problema di determinare se il linguaggio di una TM è volgare.

(a) ~~Formula questo problema come un linguaggio $GROSS_{TM}$.~~

(b) Dimostra che il linguaggio $GROSS_{TM}$ è indecidibile.

DIMOSTRA $GROSS_{TM}$ INDECIDIBILE

$$\left[\begin{array}{ccc} A_{TM} & & GROSS_{TM} \\ E_{TM} & \le_m & \text{(PROBLEMA)} \\ EQ_{TM} & & \end{array}\right]$$

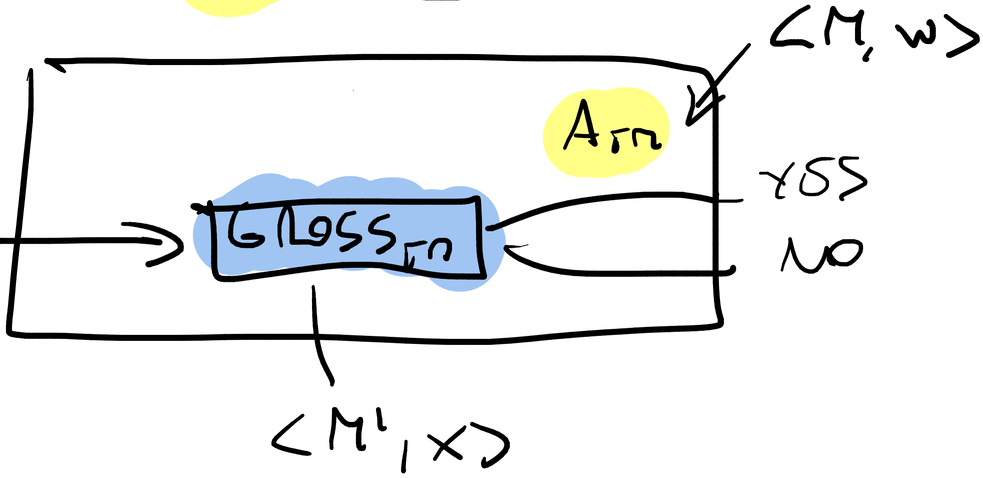
RIDUZIONE

F = FUNZ. DI RIDUZIONE

MAPPARE

$A \le_m B \Rightarrow A_{TM} \le_m GROSS_{TM}$

- RISOLVI TRAMITE B (SOTTO) IL PROBLEMA (A) (SUPER)



- NELLE CONDIZIONI DI A!

FUNZIONE DI RIDUZIONE (SE E SOLO SE FINALE)

$$[x \in A \iff f(x) \in B]$$

$A \le_m B$

(A RIDUCE B)

$\langle M, w\rangle \longrightarrow \langle M', x\rangle$

$A_{TM} = A$ $\qquad$ $GROSS_{TM} = B$

F = SU INPUT $\langle M, w\rangle$

→ 1 SIMULA M' SU INPUT x = SOTTO INPUT

→ 1.a SE w È UN GRAWLIX, M' ACCETTA

→ 1.b ALTRIMENTI, RIFIUTA / VA IN LOOP

→ 2 SE M ACCETTA, w È UN GRAWLIX

ALTRIMENTI Ø VA IN LOOP

4. **(9 punti)** Considera il seguente problema: data una TM M a nastro semi-infinito, determinare se esiste un input w su cui M sposta la testina a sinistra partendo dalla cella numero 2023 (ossia se in qualche momento durante la computazione la testina si muove dalla cella 2023 alla cella 2022).

   (a) Formula questo problema come un linguaggio 2023_{TM}.

   (b) Dimostra che il linguaggio 2023_{TM} è indecidibile.

TM

SINGOLO

NASTRO SEMI-INFINITO

[- - - - ∞]

↑

CONFINE

MULTI-NASTRO

TURING-RICONOSCIBILE = DECIDIBILE PER TM

2023_{TM} =

IF $\underbrace{w}_{\text{INPUT}}$ = "QUALCOSA" $\xrightarrow{Y}$ 2023 → 2022

$\xrightarrow{N}$ NO

LOGICA → = PROBLEMA

4. (9 punti) Considera il seguente problema: data una TM M a nastro semi-infinito, determinare se esiste un input w su cui M sposta la testina a sinistra partendo dalla cella numero 2023 (ossia se in qualche momento durante la computazione la testina si muove dalla cella 2023 alla cella 2022).

(a) Formula questo problema come un linguaggio 2023_{TM}.

(b) Dimostra che il linguaggio 2023_{TM} è indecidibile.

(a) $2023_{TM} = \{ \langle M, w \rangle$

= M è UNA TM

e w è l'input

PER CUI M SPOSTI A SX

DALLA CELLA 2023 A 2022 $\}$

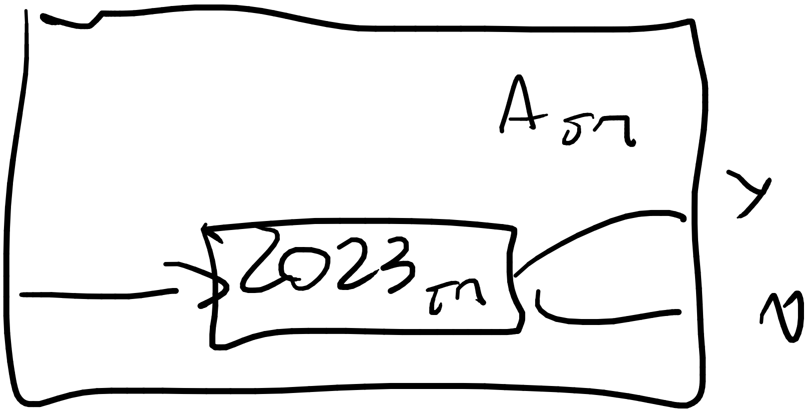
(b) 2023_{TM}

↓

$A \le_m B$

MAPPING

$A_{TM} \le_m 2023_{TM}$



F ⇒ SU INPUT $\langle M, w \rangle$:

① SIMULA M' SU INPUT X

② M SU $\underline{w}$ = ACCETTA

↓

A_{TM} DI FORMARSI SPOSTA SU 2023 A SX

X ACCETTATO DA M'

$A \leq_m B$

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

↓ ESERCIZIO

$\langle M, w \rangle \in A_{TM} \iff \langle M', x \rangle \in 2023_{TM}$

PERCHÉ F. DI RIDUZIONE FUNZIONA?

① SE $\langle M, w \rangle \in A_{TM}$

→ M' SIMULA SU INPUT x

→ SE M ACCETTA w

→ LA TM SI SPOSTA DA 2023 A 2022 CON MOVIMENTO A SX

② SE $\langle M, w \rangle \notin A_{TM}$

→ M' SIMULA SU X

→ NON RAGGIUNGE MAI LA DESTINAZIONE 2022.

2. (12 punti) Considera il linguaggio

$$\text{EVEN-HALTS} = \{\langle M\rangle \mid \text{per ogni numero naturale } n \text{ pari, } M \text{ termina la computazione su } n\}.$$

Dimostra che EVEN-HALTS è indecidibile.

1

2

< 2022 → NO DEF. LINGUAGGIO

2 EVEN-HALTS INDECIDIBILE

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

↓

$\langle M', x\rangle \in$ EVEN-HALTS

F → SU INPUT $\langle M, w\rangle$:

- M' SU INPUT X:

X è PARI ? → SI

↓

$\langle M, w\rangle$

SSE M → ACCETTA

TUTTI GLI N. PARI

(NO ALTRI MODI)

$$\left[\langle M, w\rangle \in A_{TM} \iff \langle M', x\rangle \in \text{EVEN-HALTS}\right]$$

VERIFICATORE

3. **(12 punti)** In una delle storie delle Mille e una notte, Alì Babà, mentre viaggiava con il suo asino, trovò la grotta in cui i 40 ladroni avevano nascosto il loro bottino. Come cittadino rispettoso della legge, denunciò il fatto alla polizia, ma solo dopo aver tenuto il più possibile per sé. Il problema è che c'è troppo bottino e l'asino non può portarlo tutto: c'è un limite M al peso che l'asino può trasportare. Supponiamo che ognuno degli N oggetti rubati abbia un prezzo $P[i]$ e un peso $W[i]$. Alì Babà può caricare sull'asino un numero sufficiente di oggetti in modo che il prezzo totale sia almeno L?

Formalmente, possiamo rappresentare il problema che Alì Babà deve risolvere con il linguaggio

$$ALIBABA = \left\{ \langle N, P, W, M, L\rangle \;\middle|\; \text{esiste } \underline{B \subseteq \{1, \ldots, N\}} \text{ tale che } \underline{\sum_{j\in B} W[j] \le M} \text{ e } \underline{\sum_{j\in B} P[j] \ge L} \right\}. \quad (1)\ (2)\ (3) \rightarrow V$$

(a) Dimostra che *ALIBABA* è un problema NP.

(b) Sappiamo che il linguaggio

$$SUBSET\text{-}SUM = \left\{ \langle S, t\rangle \;\middle|\; S \text{ insieme di naturali, ed esiste } S' \subseteq S \text{ tale che } \sum_{x\in S'} x = t \right\}$$

è NP-completo. Dimostra che *ALIBABA* è NP-hard, usando *SUBSET-SUM* come problema NP-hard di riferimento.

NP { PUO' RISOLVERE PROBLEMA → T. FINITO
NON È RISOLVIBILE SEMPRE }

NP → VERIFICATORE (CERTIFICATO)

RISPETTIAMO PESO MAX (NO OLTRE) $\Leftarrow (\sum W[J] \le M)$

RUBIAMO ALMENO TOT SOLDI $\Leftarrow (\sum P[J] \ge L)$

VERIFICATORE → INPUT BUONO

$V\langle N, P, W, M, L \rangle, B$

→ $B \subseteq \{1 \ldots N\}$ = VEDERE TUTTI GLI ELEMENTI

→ PESO = $\sum_{j} W$

→ PREZZO = $\sum_{j} P$

(B) ALIBABA → NP-HARD

$A \leq_m B$

↓

NP-HARD $\leq_m$ PROBLEMA

$$ALIBABA = \left\{ \langle N, P, W, M, L \rangle \;\middle|\; \text{esiste } B \subseteq \{1, \ldots, N\} \text{ tale che } \sum_{j \in B} W[j] \leq M \text{ e } \sum_{j \in B} P[j] \geq L \right\}.$$

(a) Dimostra che *ALIBABA* è un problema NP.

(b) Sappiamo che il linguaggio

$$SUBSET\text{-}SUM = \left\{ \langle S, t \rangle \;\middle|\; S \text{ insieme di naturali, ed esiste } S' \subseteq S \text{ tale che } \sum_{x \in S'} x = t \right\}$$

è NP-completo. Dimostra che *ALIBABA* è NP-hard, usando *SUBSET-SUM* come problema NP-hard di riferimento.

T = TARGET

$S \leftarrow \{3, 5, 8, 10\}$

$\sum_{x \in S'} 3 + 5 = 8 \quad S' = \{3, 5\}$

L = 8

SUBSET - SUM $\rightarrow S' \subseteq S$

$$\sum_{x \in S'} s = A$$

TARGET

ALIBABA

$\sum_{S} w \leq B$ (POSI)

T

$\sum_{S} p \geq L$ (PRESTI)

$\boxed{\text{SUBSET - SUM} \geq \boxed{\text{ALI - BABA}}}$

N $\rightarrow$ N. DI ELEMENTI

$\forall\, i \in N$

$\downarrow$

$\sum_i w[i] \leq B$ (POSI)

$\sum p[i] \geq L$

$W \leq$ TARGET $\leq P$

ISTANZA DI SUBSET-SUM

f(S,t) = ⟨N,P,W,M,L⟩ dove:
- N = n (stesso numero di elementi)
- Per ogni i = 1, ... ,n:
  * $P[i] = s_i$ (prezzo = valore elemento)
  * $W[i] = s_i$ (peso = valore elemento)
- M = t (limite peso = target)
- L = t (soglia prezzo = target)

$$\left(T \longrightarrow \sum_W , \sum_P \right)$$

ALI-BABA SUBSET-SUM

$\downarrow$

$$B = \{1 \ldots n\}$$

$$\sum_{S \in B} W[S] = T/2$$

$$\sum_{S \in B} P[S] = T/2$$

T