Cognome ______________________ **Nome** ______________________

Matricola ______________________ **Numero di CFU** ________ **Fila 1**

Università degli Studi di Bologna, Corso di Laurea in Informatica
Esame di INFORMATICA TEORICA (6 CFU), 24/05/2023

Utilizzare i riquadri bianchi per le risposte. Solo se strettamente necessario, si può allegare un foglio protocollo in coda con ulteriore testo, indicando in alto nome, cognome, fila e matricola.

Nota: nelle domande da Q2 a Q6 una risposta giusta da 1 punto, una risposta sbagliata sottrae 0.25 punti. Si puó scegliere di non rispondere, nel qual caso non vengono dati né sottratti punti.

Q1 (2 punti). Definisci (come tupla) la macchina di Turing su alfabeto di input $\{a, b\}$ che *riconosce* il linguaggio delle stringhe dove appare la lettera a (per esempio: aa, aba, $bbab$, ...). Puoi usare un diagramma per descrivere la funzione di transizione. Per il punteggio pieno, utilizza un numero minimale di stati.

Definiamo la TM come la tupla $\langle\{a, b, \sqcup\}, \{q_0, q_1\}, q_0, q_1, \delta\rangle$ dove δ é definita come segue:

$$\delta(q_0, a) = (x_1, x_2, q_1) \qquad \delta(q_0, b) = \delta(q_0, \sqcup) = (x_1, \rightarrow, q_0)$$

Qui x_1 puó essere qualsiasi simbolo in $\{a, b, \sqcup\}$, e x_2 puó essere qualsiasi azione in $\{\leftarrow, \rightarrow\}$: la scelta non cambia la definizione del linguaggio riconosciuto dalla macchina.

Q2 (4 punti). Nel seguito, sia code$(-)$ una funzione iniettiva calcolabile che codifichi maccchine di Turing come stringhe in $\{0, 1\}^*$. Per ciascuno dei seguenti linguaggi, indica se é (1) decidibile, (2) indecidibile ma riconoscibile, (3) non riconoscibile.

	Linguaggio	Decidicible	Indecidibile ma riconoscibile	Non riconoscibile
(a)	$\{y \in \{0,1\}^\star \mid y = \text{code}(M)$ per qualche TM M e M non si ferma sulla stringa vuota $\epsilon\}$			
(b)	$\{y \in \{0,1\}^\star \mid y = \text{code}(M)$ per qualche TM M e M si ferma su $\text{code}(M)\}$			
(c)	$\{y \in \{0,1\}^\star \mid y = \text{code}(M)$ per qualche TM M e M ha meno di due stati$\}$			
(d)	$\{y \in \{0,1\}^\star \mid y = \text{code}(M)$ per qualche TM M e M si ferma su tutte le stringhe di lunghezza pari$\}$			

a: (3), b: (2), c: (1), d: (3).

Q3 (4 punti). Indica (con un Si o No) a quali dei linguaggi di Q2 (indicati con (a), (b), (c) e (d)) é applicabile il teorema di Rice.

	Rice?		Rice?		Rice?		Rice?
(a)		(b)		(c)		(d)	

Il teorema di Rice si applica ad a e d.

Q4 (4 punti). Sia L un linguaggio decidibile da una macchina di Turing in tempo $T(n)$. Per quali dei seguenti quattro possibili valori di $T(n)$ possiamo dire che L é nella classe P?

	É in P?
(a) $3 + 2^n$	
(b) $n + n^3$	

	É in P?
(c) $n!$	
(d) $2^{\log_2 n}$	

(b) e (d).

Cognome ______________________		**Nome** ______________________	
Matricola ______________________		**Numero di CFU** ________	**Fila 1**

Università degli Studi di Bologna, Corso di Laurea in Informatica
Esame di INFORMATICA TEORICA (6 CFU), 24/05/2023

Utilizzare i riquadri bianchi per le risposte. Solo se strettamente necessario, si può allegare un foglio protocollo in coda con ulteriore testo, indicando in alto nome, cognome, fila e matricola.

Q5 (4 punti). Per ciascuno dei seguenti linguaggi, indica se é l'algoritmo noto di complessità minore é nella classe (1) P, (2) NP, (3) PSPACE. Si assume che $\langle - \rangle$ sia una codifica di un oggetto del problema (grafo, strategia, formula, etc.) come stringa del linguaggio. Come in classe, assumiamo che calcolare $\langle - \rangle$ impieghi tempo al piú polinomiale.

	Linguaggio	P	NP	PSPACE
(a)	Ricorda che un enunciato booleano é una formula della logica del prim'ordine (con connettivi $\wedge$, $\vee$, $\neg$) dove tutte le variabili sono vincolate da quantificatori. Considera: $\{\langle F \rangle \mid F$ é un enunciato booleano (con quantificatori) soddisfacibile$\}$			
(b)	$\{\langle F \rangle \mid F$ é una formula booleana *senza quantificatori* soddisfacibile$\}$			
(c)	Dato un grafo indiretto G, ricorda che un k-clique in G é un sottografo G' di G con k nodi, tale che ogni coppia di nodi di G' é collegata da un arco. Considera il linguaggio $\{\langle G \rangle \mid G$ ha un 5-clique$\}$			
(d)	$\{w \in \{a, b, c\}^\star \mid w$ é una stringa palindroma$\}$			

(a) PSPACE. (b) NP. (c) NP. (d) P.

Q6 (7 punti). Indica (senza dimostrazione) quali di queste affermazioni sono vere, quali sono false, e quali sono problemi aperti.

	Linguaggio	V	F	Aperto
(a)	Se L_1 e L_2 sono in NP, allora anche $L_1 \cup L_2$ é in NP.			
(b)	Esiste un problema in $PSPACE$ ma non in $NPSPACE$.			
(c)	Sia $co{-}P$ la classe dei linguaggio il cui complemento é in P. Esiste un problema in NP che non é in $co{-}P$.			
(d)	Se $L \leq HALT$, allora L é indecidibile.			
(e)	Se L é in NP, allora $L \leq_p SAT$.			
(f)	Sia L un linguaggio $PSPACE$-completo. Se $L \in NP$, allora $PSPACE \subseteq NP$.			
(g)	Alcuni linguaggi decidibili non sono in P.			

(a) Vero, ha una semplice dimostrazione diretta.

(b) Falso, il teorema di Savitch dimostra che $PSPACE = NPSPACE$.

(c) Non lo sappiamo, dal momento che $P = co - P$ e non sappiamo se $P = NP$.

(d) Falso (sarebbe vero se scrivessimo $HALT \leq L$).

(e) Vero, poiché SAT é NP-completo.

(f) Vero, per definizione di completezza.

(g) Vero, per il teorema di gerarchia di tempo.

Cognome ______________________ **Nome** ______________________

Matricola ______________________ **Numero di CFU** ________ **Fila 1**

Università degli Studi di Bologna, Corso di Laurea in Informatica
Esame di INFORMATICA TEORICA (6 CFU), 24/05/2023

Utilizzare i riquadri bianchi per le risposte. Solo se strettamente necessario, si può allegare un foglio protocollo in coda con ulteriore testo, indicando in alto nome, cognome, fila e matricola.

Q7 (5 punti). Considera i seguenti linguaggi L_1 e L_2.

$$\begin{aligned} L_1 &= \{y \in \{0,1\}^\star \mid y = \text{code}(M) \text{ per qualche TM } M \text{, e } M \text{ si ferma sulla string vuota } \epsilon\} \\ L_2 &= \{y \in \{0,1\}^\star \mid y = \text{code}(M) \text{ per qualche TM } M \text{, e } M \text{ si ferma su code}(M)\} \end{aligned}$$

Dimostra che esiste una mapping reduction da L_1 a L_2 (notazione $L_1 \leq L_2$).

Vogliamo costruire una funzione computabile f tale che $y \in L_1$ se e solo se $f(y) \in L_2$. Definiamo f come segue. Se y non é il codice di nessuna macchina di Turing, definiamo $f(y)$ come y. Altrimenti, $y = \text{code}(M)$ per qualche TM M. In questo caso, definiamo $f(y)$ come $\text{code}(M')$, dove M' é la TM definita come segue: se l'input é $\text{code}(M')$, M' simula M sul nastro vuoto. Altrimenti M' entra in un ciclo.
Prima di tutto, verifichiamo che f sia computabile. Lo é perché il problema che $y = \text{code}(M)$ per qualche TM M é decidibile, e la costruzione di M' é chiaramente definibile da un algoritmo che termina sempre.
Infine, rimane da verificare che $y \in L_1$ se e solo se $f(y) \in L_2$.

- Se $y \neq \text{code}(M)$ per nessuna TM M, allora $y \notin L_1$ e $f(y) = y \notin L_1$.
- Supponiamo ora che $y = \text{code}(M)$ per qualche TM M. Se $y = \text{code}(M) \in L_1$, allora M ferma sulla stringa vuota, quindi per definizione M' ferma su $\text{code}(M')$, quindi $M' \in L_2$.
- Altrimenti, se $y = \text{code}(M) \notin L_1$, allora M non ferma sulla stringa vuota, quindi M' non ferma su $f(\text{code}(M)) = \text{code}(M')$, quindi $M' \notin L_2$.