|  |
| --- |
| **1 - Automi**  - Dati due linguaggi L1 e L2, definire l’operazione di concatenazione L1L2, dimostrare che la concatenazione di due linguaggi regolari produce un linguaggio regolare.  - Fornire la definizione e di definire le operazioni di intersezione e complemento di linguaggi, dimostrare che i linguaggi regolari sono chiusi per le operazioni di intersezione e complemento.  - Mostrare che la classe dei linguaggi regolari risulta chiusa per l’operazione di unione. |
| **2 - Espressioni Regolari**  - Fornire la definizione ricorsiva di espressione regolare, indicando con precisione il linguaggio rappresentato.  - Sintetizzare la dimostrazione che per ogni espressione regolare esiste un NFA equivalente.  - Per ognuno dei seguenti punti dire se l'affermazione risulta vera o falsa e motivare brevemente le risposte date.  (a) Se L1 ∪ L2 è regolare, allora L1 ed L2 sono entrambi regolari.  (b) Se L è decidibile allora anche il suo complemento è decidibile.  - Enunciare il Pumping Lemma ed utilizzarlo. |
| **3/4 - MdT**  - Fornire la definizione di configurazione di una Macchina di Turing M e di stringa accettata da M.  - Fornire le definizioni (formalmente precise) di linguaggio decidibile e linguaggio Turing riconoscibile e spiegare brevemente la differenza tra le due classi di linguaggi, mostrare o confutare che i linguaggi Turing decidibili sono chiusi per l’operazione di complemento, mostrare o confutare che i linguaggi Turing riconoscibili sono chiusi per l’operazione di complemento.  - Fornire la definizione formale di Macchina di Turing deterministica multinastro.  - Fornire la settupla che definisce una macchina di Turing a 2 nastri. Dimostrare che per ogni macchina di Turing a 2 nastri esiste una macchina di Turing equivalente a singolo nastro. |
| **5 - Decidibilità**  - Fornire la definizione di insieme numerabile.  - Mostrare che l’insieme di tutte le coppie (i; j) dove i e j sono numeri interi con i < j risulta numerabile.  - Utilizzare il metodo della diagonalizzazione per mostrare che l’insieme {x | x ´e un numero reale t.c. 0 < x < 1} non è numerabile.  - Dimostrare (mediante diagonalizzazione) l’esistenza di linguaggi che non sono Turing riconoscibili.  - Dimostrare l’esistenza di linguaggi che non sono Turing riconoscibili. |
| **6 - Riducibilità**  - Dare la definizione di riduzione e di linguaggio indecidibile.  - Mostrare la riduzione ATM ≤m HALTTM, dedurne che HALTTM è indecidibile.  - Definire il linguaggio EQTM.  - Dimostrare che EQTM non è Turing riconoscibile e non è co-Turing riconoscibile. Occorre enunciare con precisione tutti i risultati intermedi utilizzati.  - Enunciare il teorema di Rice, è possibile utilizzarlo per mostrare che il seguente linguaggio risulta indecidibile? Giustificare la risposta  L={⟨M⟩| M ´e una MdT che accetta ogni input di lunghezza pari}. |
| **7/8 - Complessità**  - Fornire le definizioni delle classi P, EXP, NP, co-NP, e la classe dei problemi NP-completi.  - Definire in maniera formale e rigorosa la classe NP ed il concetto di problema NP-completo.  - Illustrare il concetto di Self-reducibility  - Illustrare i concetti di riduzione polinomiale e di Self-reducibility.  - Fornire un diagramma che illustra le relazioni che si sa/suppone esistere tra queste classi; giustificando le relazioni indicate (P=NP e P≠NP).  - Siano X e Y problemi di decisione. Si sa che X ≤*P* Y. Per ognuna delle seguenti affermazioni dire se essa è:  (a) Se Y è NP-completo lo è anche X.  (b) Se X è NP-completo lo è anche Y.  (c) Se Y è NP-completo e X è in NP allora X è NP-complete.  (d) Se X è NP-completo e Y è in NP allora Y è NP-complete.  (e) X e Y non possono essere entrambi NP-completi.  (f) Se X è in P, allora Y è in P.  (g) Se Y è in P, allora X è in P.  - Si considerino 4 problemi A, B, C e D. Ognuno può appartenere o meno alla classe NP. Si conosce l’esistenza delle seguenti riduzioni:  A ≤P B, B ≤P C, D ≤P C.  Per ognuna delle affermazioni seguenti indicare se è sicuramente VERA, sicuramente FALSA oppure NON SI SA (cioè dipende dai problemi e dalla relazione tra le classi P e NP); giustificare brevemente le risposte.  • Se A è NP-completo allora C è NP-completo.  • A è NP-completo e C ∈ P.  • Se A è NP-completo e B ∈ NP, allora B è NP-completo.  • Se C è NP-completo allora D ∈ NP.  - Mostrare in maniera formale e rigorosa le seguenti inclusioni tra classi di complessità, enunciando in maniera precisa eventuali risultati intermedi utilizzati:  1) P *⊆* NP  2) NP *⊆* EXP  3) P *⊆* co *−* NP  - Definire i problemi di decisione 3-SAT e INDEPENDENT–SET.  - Si descriva e commenti l’istanza di INDEPENDENT–SET nella riduzione 3-SAT≤*P*INDEPENDENT–SET.  - Definire i linguaggi 3-SAT e VERTEX-COVER (occorre definire ogni termine utilizzato nella definizione).  - Si descriva l’istanza di VERTEX-COVER nella riduzione polinomiale di 3-SAT a VERTEX-COVER.  - Si definisca il problema di decisione SAT.  - Definire il problema di decisione SUBSET – SUM, HAM-CYCLE e TSP.  - Data la seguente istanza di 3−SAT si descriva l’istanza di SUBSET −SUM nella riduzione polinomiale di 3 − SAT a SUBSET − SUM |