Esercizio 5.22

Dimostra che A è Turing-riconoscibile se e solo se A $\leq_m$ A_TM.

Dimostrazione:

(⇒) Se A è Turing-riconoscibile, allora A $\leq_m$ A_TM:

Sia M una macchina di Turing (MT) che riconosce A. Costruiamo una riduzione f da A a A_TM come segue:

Per ogni input w, definiamo f(w) = ⟨M, w⟩, dove M è la MT che riconosce A.

Questa funzione f è chiaramente calcolabile totale, poiché la codifica ⟨M, w⟩ può essere costruita meccanicamente per qualsiasi w. Verifichiamo che si tratta di una riduzione valida:

- Se w ∈ A, allora M accetta w, quindi ⟨M, w⟩ ∈ A_TM.
- Se w ∉ A, allora M non accetta w, quindi ⟨M, w⟩ ∉ A_TM.

Quindi, w ∈ A se e solo se f(w) ∈ A_TM, provando che A $\leq_m$ A_TM.

Nota: In alternativa, si potrebbe anche definire una riduzione usando una MT M_w specifica per ogni w, come segue: f(w) = ⟨M_w, w⟩, dove M_w è una MT che:

1. Su qualsiasi input, ignora l'input e simula M su w
2. Accetta se e solo se M accetta w

Anche questa riduzione è calcolabile totale e soddisfa le condizioni richieste.

(⇐) Se A $\leq_m$ A_TM, allora A è Turing-riconoscibile:

Supponiamo che A $\leq_m$ A_TM tramite una riduzione calcolabile f. Sappiamo che A_TM è Turing-riconoscibile. Costruiamo una MT R che riconosce A:

Per ogni input w, R opera come segue:

1. Calcola f(w) = ⟨M, x⟩ per qualche MT M e stringa x
2. Simula una MT riconoscitrice per A_TM su ⟨M, x⟩
3. Se la MT per A_TM accetta, allora R accetta; altrimenti R continua a simulare

Dato che f è calcolabile e A_TM è Turing-riconoscibile, R è una MT valida che riconosce A. Pertanto, A è Turing-riconoscibile.

Esercizio 5.23

Dimostra che A è decidibile se e solo se A $\leq_m$ 0*1*.

Dimostrazione:

(⇒) Se A è decidibile, allora A $\leq_m$ 0*1*:

Sia M una MT che decide A. Costruiamo una riduzione f da A a 0*1* come segue:

Per ogni input w, definiamo:

- f(w) = 0 se w ∈ A
- f(w) = 10 se w ∉ A

Questa funzione f è calcolabile totale perché M decide A in tempo finito. Verifichiamo che si tratta di una riduzione valida:

- Se w ∈ A, allora f(w) = 0 ∈ 0*1* (è nella forma 0^n1^m con n=1, m=0)
- Se w ∉ A, allora f(w) = 10 ∉ 0*1* (poiché in 0*1* tutti gli 0 precedono tutti gli 1, mentre in "10" un 1 precede uno 0)

Quindi A $\leq_m$ 0*1*.

Nota: Il linguaggio 0*1* è un linguaggio regolare che contiene tutte e sole le stringhe dove tutti gli 0 precedono tutti gli 1, incluse le stringhe composte solo da 0 o solo da 1, o la stringa vuota.

(⇐) Se A $\leq_m$ 0*1*, allora A è decidibile:

Supponiamo che A $\leq_m$ 0*1* tramite una riduzione calcolabile f. Poiché 0*1* è decidibile (è un linguaggio regolare), esiste una MT D che decide 0*1*. Costruiamo una MT M che decide A:

Per ogni input w, M opera come segue:

1. Calcola f(w)
2. Esegue D su f(w)
3. Se D accetta f(w), allora M accetta w; altrimenti M rifiuta w

Dato che f è calcolabile e D decide 0*1*, M è una MT valida che decide A. Pertanto, A è decidibile.

Esercizio 5.24

Sia J = {w| o w = 0x per qualche x ∈ A_TM, o w = 1y per qualche y ∈ Ā_TM}. Dimostra che né J né J̄ è Turing-riconoscibile.

Dimostrazione:

J non è Turing-riconoscibile:

Supponiamo per assurdo che J sia Turing-riconoscibile. Allora esiste una MT R che riconosce J.

Costruiamo una MT M che riconoscerebbe $\overline{A}_{TM}$ come segue: Per ogni input $\langle M', w\rangle$, M opera come segue:

1. Forma la stringa $s = 1\langle M', w\rangle$
2. Esegue R su s
3. Se R accetta s, allora M accetta; altrimenti M continua a simulare

Analizziamo il comportamento di M:

- Se $\langle M', w\rangle \in \overline{A}_{TM}$, allora $s = 1\langle M', w\rangle \in J$ per definizione di J, quindi R accetta s, e M accetta.
- Se $\langle M', w\rangle \notin \overline{A}_{TM}$, allora $s = 1\langle M', w\rangle \notin J$, quindi R non accetta s, e M non accetta.

Ma questo significa che M riconoscerebbe $\overline{A}_{TM}$, che è in contraddizione con il fatto noto che $\overline{A}_{TM}$ non è Turing-riconoscibile (per il Teorema di Rice). Quindi, J non può essere Turing-riconoscibile.

$\overline{J}$ non è Turing-riconoscibile:

Supponiamo per assurdo che $\overline{J}$ sia Turing-riconoscibile. Allora esiste una MT R' che riconosce $\overline{J}$.

Costruiamo una MT M' che riconoscerebbe $\overline{A}_{TM}$ come segue: Per ogni input $\langle M', w\rangle$, M' opera come segue:

1. Forma la stringa $s = 0\langle M', w\rangle$
2. Esegue R' su s
3. Se R' accetta s, allora M' accetta; altrimenti M' continua a simulare

Analizziamo il comportamento di M':

- Se $\langle M', w\rangle \notin A_{TM}$ (quindi $\langle M', w\rangle \in \overline{A}_{TM}$), allora $s = 0\langle M', w\rangle \notin J$ (poiché la prima condizione della definizione di J non è soddisfatta), quindi $s \in \overline{J}$, quindi R' accetta s, e M' accetta.
- Se $\langle M', w\rangle \in A_{TM}$ (quindi $\langle M', w\rangle \notin \overline{A}_{TM}$), allora $s = 0\langle M', w\rangle \in J$, quindi $s \notin \overline{J}$, quindi R' non accetta s, e M' non accetta.

Pertanto, M' accetta $\langle M', w\rangle$ se e solo se $\langle M', w\rangle \in \overline{A}_{TM}$, cioè M' riconoscerebbe $\overline{A}_{TM}$, che sappiamo non essere Turing-riconoscibile. Questa è una contraddizione.

Pertanto, né J né $\overline{J}$ è Turing-riconoscibile.

Esercizio 5.25

Fornisci un esempio di un linguaggio indecidibile B, dove $B \leq_m \bar{B}$.

Soluzione:

Definiremo il linguaggio K = {⟨M, w⟩ | M si ferma su input w con un numero dispari di passi}.

Primo: Dimostriamo che K è indecidibile.

Mostriamo questo riducendo il problema dell'arresto (HALT) a K. Ricordiamo che HALT = {⟨M, w⟩ | M si ferma su input w} è indecidibile.

Definiamo una riduzione f: HALT → K come segue: Per ogni ⟨M, w⟩, costruiamo f(⟨M, w⟩) = ⟨M', w⟩ dove M' è una MT che:

1. Simula M su w
2. Se M si ferma, esegue un passo aggiuntivo e poi si ferma

Questa funzione f è calcolabile totale poiché possiamo costruire meccanicamente M' a partire da M. Verifichiamo che sia una riduzione valida:

- Se ⟨M, w⟩ ∈ HALT, allora M si ferma su w dopo un certo numero di passi, diciamo n. Allora M' si fermerà dopo n+1 passi, che è un numero dispari se n è pari, e un numero pari se n è dispari. In questo caso, ⟨M', w⟩ ∈ K se e solo se n è pari.
- Se ⟨M, w⟩ ∉ HALT, allora M non si ferma su w, quindi neanche M' si fermerà su w, e ⟨M', w⟩ ∉ K.

Questo sembra creare un problema perché la riduzione non preserva l'appartenenza. Modifichiamo leggermente la riduzione:

Per ogni ⟨M, w⟩, costruiamo f(⟨M, w⟩) = ⟨M'', w⟩ dove M'' è una MT che:

1. Esegue un passo fittizio (che non fa nulla ma conta come un passo)
2. Simula M su w

Ora:

- Se ⟨M, w⟩ ∈ HALT, allora M si ferma su w dopo n passi. Quindi M'' si fermerà dopo n+1 passi, che è sempre un numero dispari se n è pari, e pari se n è dispari.
- Se ⟨M, w⟩ ∉ HALT, allora M non si ferma su w, quindi neanche M'' si fermerà su w, e ⟨M'', w⟩ ∉ K.

Tuttavia, questo non è ancora sufficiente. Costruiamo una riduzione migliore:

Per ogni ⟨M, w⟩, definiamo f(⟨M, w⟩) = ⟨M''', w⟩ dove M''' è una MT che:

1. Simula M su w
2. Se M si ferma, allora M''' si ferma immediatamente

Ora:

- Se $\langle M, w\rangle \in$ HALT, allora M''' si ferma su w esattamente con lo stesso numero di passi di M, quindi $\langle M''', w\rangle \in K$ se e solo se M si ferma con un numero dispari di passi.
- Se $\langle M, w\rangle \notin$ HALT, allora M''' non si ferma su w, quindi $\langle M''', w\rangle \notin K$.

Questo non preserva l'appartenenza per ogni istanza. Fortunatamente esiste una terza costruzione più diretta:

Definiamo $K\_0 = \{\langle M, w\rangle \mid M$ si ferma su input $w\}$.

K_0 è indecidibile (è semplicemente il problema dell'arresto). Ora definiamo: $K = \{\langle M, w, b\rangle \mid$ (M si ferma su w e b=0) o (M non si ferma su w e b=1)$\}$

K è chiaramente indecidibile, poiché decidere K permetterebbe di decidere K_0.

Secondo: Dimostriamo che $K \leq_m \bar{K}$.

Definiamo la riduzione g da K a $\bar{K}$ come segue: Per ogni $\langle M, w, b\rangle$, definiamo $g(\langle M, w, b\rangle) = \langle M, w, 1-b\rangle$

Questa funzione g è chiaramente calcolabile totale. Verifichiamo che è una riduzione valida:

- Se $\langle M, w, b\rangle \in K$, allora o (M si ferma su w e b=0) o (M non si ferma su w e b=1)
  - Se M si ferma su w e b=0, allora $g(\langle M, w, b\rangle) = \langle M, w, 1\rangle$, che non è in K perché M si ferma ma b=1. Quindi $\langle M, w, 1\rangle \in \bar{K}$.
  - Se M non si ferma su w e b=1, allora $g(\langle M, w, b\rangle) = \langle M, w, 0\rangle$, che non è in K perché M non si ferma ma b=0. Quindi $\langle M, w, 0\rangle \in \bar{K}$.
- Se $\langle M, w, b\rangle \notin K$, allora o (M si ferma su w e b=1) o (M non si ferma su w e b=0)
  - Se M si ferma su w e b=1, allora $g(\langle M, w, b\rangle) = \langle M, w, 0\rangle$, che è in K perché M si ferma e b=0. Quindi $\langle M, w, 0\rangle \notin \bar{K}$.
  - Se M non si ferma su w e b=0, allora $g(\langle M, w, b\rangle) = \langle M, w, 1\rangle$, che è in K perché M non si ferma e b=1. Quindi $\langle M, w, 1\rangle \notin \bar{K}$.

In tutti i casi, $\langle M, w, b\rangle \in K$ se e solo se $g(\langle M, w, b\rangle) \in \bar{K}$, dimostrando che $K \leq_m \bar{K}$.

Quindi, K è un linguaggio indecidibile tale che $K \leq_m \bar{K}$.