1. Si consideri il problema di stabilire se una TM è tale da accettare un linguaggio infinito, formalmente il problema è descritto dal seguente linguaggio:

   INF_{TM} = {<T> | T è una TM e L(T) è infinito}.

Si costruisca una riduzione basata su una funzione da A_{TM} a INF_{TM}, per dimostrare la non decidibilità di INF_{TM} .
Questa riduzione ci permette di dedurre qualcosa circa la decidibilità o la Turing riconoscibilità di $notINF_{TM}$?

Sol. Una riduzione da A_{TM} a INF_{TM} deve associare a istanze di A_{TM} istanze di INF_{TM} in modo tale che a istanze sì di A_{TM} corrispondano istanze sì di INF_{TM} e a istanze no di A_{TM} istanze no di INF_{TM}.
Un'istanza di A_{TM} è una coppia formata da una TM e da un suo input, <M,w>, e dobbiamo <u>costruire</u> un'istanza del problema dell'infinito da associarvi, cioè un TM T, in modo tale che L(T) è infinito sse w è in L(M). Dobbiamo fare in modo che il comportamento della TM T dipenda da quello di M su w. Definiamo T in modo che accetti ogni parola dell'alfabeto input se M accetta w e non accetti niente in caso contrario. In questo modo L(T) = Σ* se w è in L(M) e L(T) = ∅ altrimenti. Per ottenere questo risultato dobbiamo definire la TM R che calcola la funzione di riduzione come segue:

R: input <M,w>
Output <T>
T è così costruita:
T: input x
esegui M su w
se M accetta w accetta x
se M rifiuta w rifiuta x

Controlliamo la correttezza di R:
Se M accetta w allora T accetta un suo qualsiasi input e quindi L(T) = Σ*, dove Σ è l'alfabeto di input di T. Se viceversa M non accetta w, allora si danno due casi: o M si ferma e rifiuta w o M non si ferma su w, ma in entrambi questi casi T non può accettare niente e quindi L(T) = ∅ che è finito.
La non decidibilità di INF_{TM} deriva dalla riduzione perché se disponessimo di una TM che decide INF_{TM} allora combinando in sequenza la TM R che calcola la riduzione e questa ipotetica TM per INF_{TM} otterremmo una TM che decide A_{TM}.
Poiché $A_{TM} \leq_m INF_{TM}$ è equivalente a $notA_{TM} \leq_m notINF_{TM}$ possiamo concludere che il complemento di INF_{TM} non è Turing riconoscibile, perchè se lo fosse potrei dedurne che not A_{TM} è Turing riconoscibile.

2. Si consideri il problema di stabilire se una TM accetta un linguaggio regolare, formalmente il problema è descritto dal seguente linguaggio:

Reg_{TM} = {<T> | T è una TM e L(T) è regolare}.

a. Si costruisca una riduzione basata su una funzione da A_{TM} a Reg_{TM}, per dimostrare la non decidibilità di Reg_{TM} .
b. Questa riduzione ci permette di dedurre qualcosa circa la decidibilità o la Turing riconoscibilità di $notReg_{TM}$?

Per a. si vedano gli appunti già messi in rete.
Per b. vale quanto affermato per INF_{TM}, quindi $notReg_{TM}$ è non Turing riconoscibile.

3. Resta da chiedersi se Reg_{TM} è o no Turing riconoscibile. Per dimostrare che Reg_{TM} non è Turing riconoscibile si dovrebbe costruire una riduzione da not A_{TM} a Reg_{TM}, o equivalentemente da A_{TM} a $notReg_{TM}$.
Una riduzione da A_{TM} a $notReg_{TM}$ deve associare a istanze di A_{TM} istanze di $notReg_{TM}$ in modo tale che a istanze sì di A_{TM} corrispondano istanze sì di $notReg_{TM}$ e a istanze no di A_{TM} istanze no di $notReg_{TM}$.
Un'istanza di A_{TM} è una coppia formata da una TM e da un suo input, <M,w>, e dobbiamo <u>costruire</u> un'istanza del problema della non regolarità da associarvi, cioè una TM T, in modo tale che L(T) non è regolare sse w è in L(M). In questo caso possiamo usare il linguaggio vuoto come istanza no, visto che è regolare. Come istanza sì, invece prendiamo il solito $\{0^n1^n \mid n\geq 0\}$.

La TM R è così definita:
input <M,w>
output: la TM T qui definita:
T = input x
    esegui M su w e
    se M accetta w,
    se $x = 0^n1^n$, per qualche n, allora accetta x altrimenti rifiuta
    se M rifiuta w, rifiuta x.

E' evidente che se w è accettata da M allora L(T) = $\{0^n1^n \mid n\geq 0\}$, mentre se M non è accettata da M L(T)= ∅, sia nel caso che M non si fermi su w sia nel caso che M rifiuti w. Quindi abbiamo associato un linguaggio regolare, il linguaggio vuoto a un'istanza no del problema dell'appartenenza e un linguaggio non regolare a un'istanza sì del problema, come si voleva.

4. Se $L \leq_m A_{TM}$ è possibile che $notA_{TM} \leq_m L$?

Sol. NO, perchè se $L \leq_m A_{TM}$ allora L è Turing riconoscibile e quindi $notA_{TM}$ non può ridursi a L, perchè la riduzione implicherebbe che L non è Turing riconoscibile.

5. Se $L_1 \leq_m L_2$ cosa possiamo dire se

a. L_2 è decidibile e non si sa nulla su L_1
b. L_1 è decidibile e non si sa nulla su L_2
c. L_2 è Turing riconoscibile e non si sa nulla su L_1
d. L_1 è Turing riconoscibile e non si sa nulla su L_2
e. non si sa nulla su L_1 né su L_2

Sol.

a. L_1 è decidibile infatti un algoritmo per decidere L_1 si ottiene da quello per L_2 messo in sequenza di quello che calcola la riduzione.
b. La riduzione e l'ipotesi non ci consente di dedurre alcunché su L_2.
c. Come nel caso a. una TM che Turing riconosce L_1 si ottiene da quella per L_1 componendo in sequenza la TM che calcola la riduzione e quella per L_2.
d. Come nel caso b, non possiamo concludere alcunché.
e. Anche in questo caso non possiamo concludere alcunché.

6. Si dimostri che L={<T> | T è una TM e L(T) accetta ε } è Turing riconoscibile

Sol. Dobbiamo definire una riduzione $L \leq_m A_{TM}$. Una riduzione da L a A_{TM} deve associare a istanze di L a istanze di A_{TM} in modo tale che a istanze sì di L corrispondano istanze sì di A_{TM} e a istanze no di L istanze no di A_{TM}. Vista la definizione di L potremmo associare a <T> la coppia <T,ε>, infatti se <T> è in L, cioè è un'istanza sì, allora T accetta ε, ma allora <T,ε> è un'istanza sì di A_{TM}. Mentre se <T> non è in L e quindi T non accetta ε è evidente che <T,ε> è un'istanza non di di A_{TM}.

7. Si dimostri che E_{TM}={<T> T è una TM | L(T) =∅} non è Turing riconoscibile.

Sol. Dobbiamo definire una riduzione $notA_{TM} \leq_m E_{TM}$, o più facilmente quella equivalente $A_{TM} \leq_m notE_{TM}$. Una riduzione deve associare a istanze di A_{TM} istanze di $notE_{TM}$ in modo tale che a istanze sì di A_{TM} corrispondano istanze sì di $notE_{TM}$ e a istanze no di da A_{TM} istanze no di $notE_{TM}$. Questo vuole dire che la TM T che dobbiamo associare a un istanza <M,w> deve essere tale che w è in L(M) sse L(T)≠∅.

Questo non è difficile, possiamo usare la stessa riduzione usata per il problema dell'infinito:

R: input <M,w>
Output <T>
La codifica della TM T è basata sulle seguenti definizioni delle due TM:
T: input x
esegui M su w
se M accetta w accetta x
se M rifiuta w rifiuta x

Infatti se M accetta w allora $L(T) = \Sigma^*$ se w è in L(M) e $L(T) = \varnothing$ altrimenti.
Quindi $L(T) \neq \varnothing$se M accetta w mentre $L(T) = \varnothing$ altrimenti.

8. Si dimostri che L= { <T> | T è una TM e |L(T)|>3} non è decidibile.

Sol. Dobbiamo definire una riduzione $A_{TM} \leq_m L$. Una riduzione da A_{TM} a L deve associare istanze di A_{TM} a istanze di L in modo tale che a istanze sì di A_{TM} corrispondano istanze sì di L e a istanze no di A_{TM} istanze no di L.

Qui ancora si può usare la stessa riduzione usata per per il problema dell'infinito:

R: input <M,w>
Output <T>
La codifica della TM T è basata sulle seguenti definizioni delle due TM:
T: input x
esegui M su w
se M accetta w accetta x
se M rifiuta w rifiuta x

Infatti se M accetta w allora $L(T) = \Sigma^*$ e quindi $|L(T)| > 3$, se w è in L(M) e $L(T) = \varnothing$ altrimenti.
Quindi $|L(T)| > 3$ se M accetta w mentre $|L(T)| \leq 3$ altrimenti.

Questa riduzione dice anche che $\mathrm{not}A_{TM} \leq_m \mathrm{not}L$ e quindi che il problema di determinare se una TM T accetta al più 3 parole non è Turing riconoscibile.

Resta da determinare se L è Turing riconoscibile, per farlo dobbiamo costruire una riduzione $L \leq_m A_{TM}$. Una riduzione da L a A_{TM} deve associare a istanze di L istanze di A_{TM} in modo tale che a istanze sì di L corrispondano istanze sì di A_{TM} e a istanze no di L istanze no di A_{TM}. Dovremmo associare a <T> una coppia <T',w>, in modo tale che <T> è in L, cioè è un'istanza sì, allora T' accetta w. Qui la riduzione è più complicata perchè bisogna controllare se T accetta più di tre parole. Per farlo bisogna eseguire T su un suo input fino a contarne 4, ma bisogna evitare di eseguire T su un input sul quale non si ferma. Limitiamo allora il numero di passi che T può eseguire sul suo input, sfruttando l'input di T', che è un parola qualunque.

R: input <T>
Output <T', ε >
La TM T' è a due nastri ed è descritta di seguito:
T': input x
genera la parola vuota sul secondo nastro e sia y = ε
azzera un contatore
1. esegui T su y
   se T accetta y in |x| passi incrementa il contatore
   se il contatore è uguale a 4 accetta ε, altrimenti o se T rifiuta genera sul secondo nastro la parola successiva, mettila in y e torna al punto 1.

La TM T' accetta ε se |L(T)| > 3, mentre T' non si ferma se T non accetta più di 3 parole, infatti in tal caso continua a generare nuovi input su cui verificare il comportamento di T.
Per convincersi di questo, notiamo che se una parola w è in L(T) allora sarà accettata in k passi, ma allora T' sul primo input x tale che |x| = k incrementa il contatore.