Raccolta domande di tutte le lezioni di Wooclap:

1)

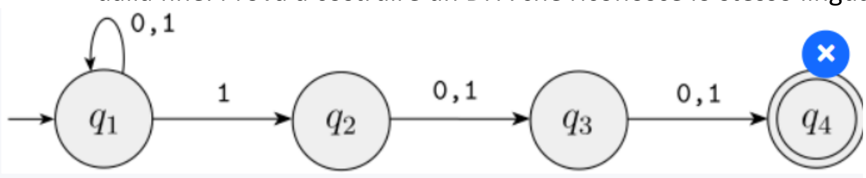
- Quali sono gli input della funzione di transizione di un DFA? → Uno stato e un simbolo
- Un DFA deve avere un solo stato finale? → Falso
- Qual è l'input del problema "È un numero primo?" → {0,1,2,3,4,5..}
- Quale insieme di stringhe rappresenta il problema nel caso "È un numero primo?" → {2,3,5,7,11,13...}
- Per ogni elemento, stabilire se è carattere, parola, linguaggio:

1) a → carattere
2) abracadabra → parola
3) {abracadabra, apostrofo, a} → linguaggio di tre parole
4) {abracadabra} → linguaggio composto da una parola

- Dato l'alfabeto ∑ = {0,1} quante sono le stringhe appartenenti al linguaggio? → 16
- Quale dei seguenti linguaggi sull'alfabeto {0,1} contiene un numero infinito di stringhe? → Tutte le stringhe che iniziano con 1
- Dato l'alfabeto ∑ = {0,1} quante sono le stringhe contenute in Σ^0? → ε
- Quante stringhe ci sono nel linguaggio sull'alfabeto {0,1} di tutte le stringhe di lunghezza *n*? → 2^n
- Un DFA possiede una quantità di memoria molto limitata → Vero
- L'insieme di tutti gli stati di un automa si indica con → Q
- Quali sono gli input della funzione di transizione di un DFA? → Uno stato e un simbolo
- Un DFA deve avere un solo stato finale → Falso

2)

1) L'NFA mostrato nella figura riconosce il linguaggio di tutte le stringhe con un 1 nella terza posizione dalla fine. Prova a costruire un DFA che riconosce lo stesso linguaggio: quanti stati possiede?



Risposta: Si risponde costruendo tutte le transizioni da tutte le combinazioni stati quindi *2^4 = 16*
Per capire gli stati effettivamente utili non si fa la tabella di transizione (con cui si trovano tutti gli stati, quindi al completo), ma il diagramma di transizione. Qui gli stati utili sono 8.

2) Qual è l'OUTPUT della funzione di transizione di un NFA?
   Risposta: Un insieme di stati

3) In un NFA, quante transizioni con lo stesso simbolo possiamo avere in uno stato?
   Risposta: Un numero a piacere, zero compreso.

4) Come si rappresenta la computazione di un NFA?
   Risposta: Con un albero di possibili transizioni
5) Un NFA accetta una stringa quando…?
   Risposta: Esiste una computazione che termina in uno stato finale

6) Un NFA rifiuta una stringa quando…?
   Risposta: Nessuna computazione termina in uno stato finale

3)

1) Una epsilon-transizione è:
   Risposta: Una transizione che non consuma l'input

2) NFA ed ε-NFA riconoscono la stessa classe di linguaggi?
   Risposta: Vero

3) Siano *L* e *M* linguaggi regolari. Per ognuna delle operazioni seguenti, indica se puoi costruire un automa che riconosce il linguaggio. Puoi scegliere più di una risposta.
   Risposte (qui a destra listate):

Unione $L \cup M$

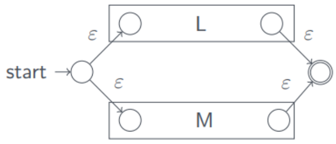
Concatenazione $L.M$

Star L^*

Intersezione $L \cap M$

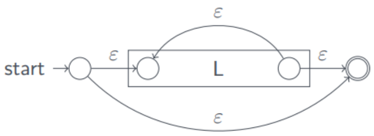
Ecco esempi possibili di automi (risposta quindi anche al file di esercizi "*03-esercizi*").

- $\mathsf{L + M}$



- L.M

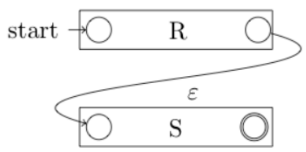
- $\mathsf{L^*}$



4)

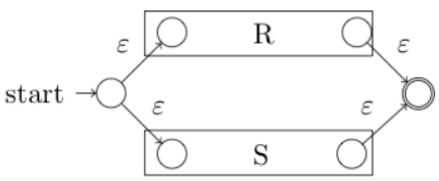
Sappiamo che R e S sono due automi a stati finiti.
Associate ogni figura al linguaggio riconosciuto dall'automa.

In queste risposte sono nell'ordine: concatenazione (sx), unione (dx) e chiusura di Kleene/star (sotto).



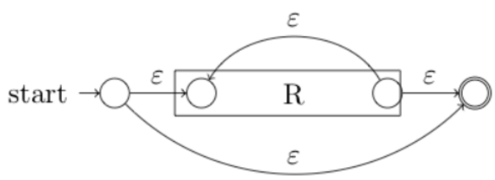
A. $L(R).L(S)$

A. $L(R).L(S)$



C. $L(R)^*$

B. $L(R) \cup L(S)$



B. $L(R) \cup L(S)$

C. $L(R)^*$

Risposta: Unione L U M

Teorema: Se L è un linguaggio regolare, allora anche $\overline{L}$ è un linguaggio regolare.

DIMOSTRAZIONE:
Sia $A = (Q, \Sigma, \delta, q_0, F)$ un NFA che accetta il linguaggio L. Allora se scambio stati finali con stati non finali ottengo l'NFA $\overline{A} = (Q, \Sigma, \delta, q_0, Q - F)$ che riconosce il linguaggio $\overline{L}$

DOMANDA: è giusta questa dimostrazione?

Risposta:
L'automa è nondeterministico, rappresentabile da un albero di computazione.
In questo caso, scambiare stati finali con non finali funziona lo stesso; l'osservazione funziona solo se abbiamo un DFA con una sola computazione (rappresentabile una lista linkata).
Infatti un DFA sceglie una sola strada per finire; NFA ha il nondeterminismo, quindi può scegliere più strade e non è più verificata la proprietà di univocità. Ragionevolmente quindi se la domanda fosse stata rivolta ad un DFA, la risposta sarebbe stata SI per questo motivo.

Risposta corretta: NO, non è corretta

Quali espressioni regolari possono generare la stringa "ababout"? Puoi scegliere più di una risposta.
Risposte:
- (ab)*out
- (a + b)*out

Quali espressioni regolari possono generare la stringa "abbbout"? Puoi scegliere più di una risposta.
Risposte:
- (ab)*out
- (a + b)*out

Quali espressioni regolari possono generare la stringa "bbbbbbout"? Puoi scegliere più di una risposta.
Risposta:
- (a+b)*out

Quali espressioni regolari possono generare la stringa "out"? Puoi scegliere più di una risposta.
Risposte:
- (ab)*out
- (ab)* + out
- (a + b)*out

Costruisci una ER sull'alfabeto $\{0, 1\}$ che rappresenti tutte le stringhe che contengono la sottostringa 101

Risposte:
- (0+1)*101(0+1)*
- (0+1)*(101)(0+1)*

Costruisci una ER sull'alfabeto $\{a, b, c\}$ che rappresenti tutte le stringhe la cui lunghezza è un multiplo di 3
Risposta:
- ((A+B+C)(A+B+C)(A+B+C))*

Supponi di modificare la costruzione che abbiamo usato per fare l'unione di linguaggi regolari in modo che produca l'automa

$$A = (Q_L \times Q_M, \Sigma, \delta, (q_L, q_M), F)$$

dove

$$\delta((p,q),a) = (\delta_1(p,a), \delta_2(q,a)) \text{ (come prima)}$$

ma

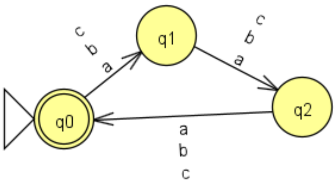
$$F = \{(p,q) \mid p \in F_1 \text{ o } q \in F_2\},$$

cioè F è l'insieme delle coppie dove l'uno o l'altro elemento è uno stato finale.

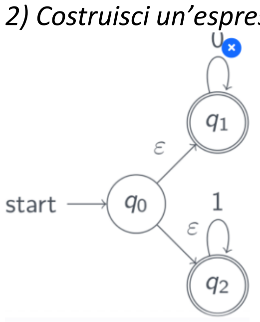
Cosa ci darebbe questa costruzione?

5)

1) *Costruisci un automa equivalente all'espressione regolare ((a+b+c)(a+b+c)(a+b+c))* (puoi caricare un'immagine con la soluzione)*

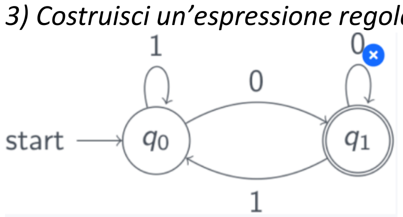


2) Costruisci un'espressione regolare equivalente all'automa in figura:



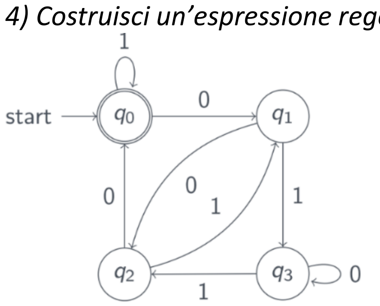
0*+1*

3) Costruisci un'espressione regolare equivalente all'automa in figura:



10(1*00*)*

4) Costruisci un'espressione regolare equivalente all'automa in figura:



(1+0(10*1+0)(1(10*1+0))*0)*

6)

Quanti stati ci sono nell'automa che riconosce il linguaggio $\{0^n 1^n \mid n \geq 0\}$?

La rappresentazione condurrebbe ad un numero infinito di stati e non sarebbe più un automa a stati finiti.
Risposta corretta: Infiniti

Il linguaggio $\{0^n 1^n \mid n \geq 0\}$ è regolare?

Si intende per “regolare” un linguaggio riconoscibile da un automa a stati finiti che lo riconosce.
Questo non è il caso.
Risposta corretta: NO
Attenzione: l’elevazione a potenza ad “n” non è infatti permessa nei linguaggi regolari.

Per applicare il Pumping Lemma ad un linguaggio, consideriamo una stringa w che appartiene a L di lunghezza $\geq k$ e la spezziamo in _________ parti

Risposta: 3

Se selezioniamo una stringa w tale che $w \in L$ e $w = xyz$, quale delle seguenti parti non può essere una stringa vuota?

Risposta: *y*

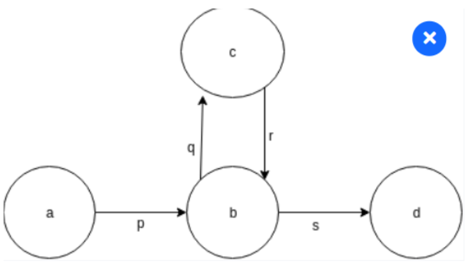
Dato un linguaggio L e una stringa w tale che $w \in L$, $w = xyz$ e $|w| \geq k$ per un numero intero costante k. Quale può essere la lunghezza massima della sottostringa xy, ossia $|xy| \leq ?$

Risposta: *k*

Rispondi in conformità con la terza e ultima affermazione del Pumping Lemma:
Per ogni _______ $xy^i z \in L$

Risposta: i >= 0

Se d è uno stato finale, quale delle seguenti affermazioni è corretta secondo il diagramma?



Risposta:

$x = p, y = qr, z = s$

Quali delle seguenti affermazioni sono vere? (puoi scegliere più di una risposta)

Risposte:
- Se un linguaggio è regolare, allora rispetta le condizioni del Pumping Lemma
- Se un linguaggio NON rispetta le condizioni del Pumping Lemma, allora NON è regolare

Possiamo usare il Pumping Lemma per (puoi scegliere più di una risposta):

Risposte:
Se dimostro L non rispetta P.L. (pumping lemma) → L non è regolare
Se dimostro che L rispetta P.L. → Non posso dire niente
Per dimostrare che è regolare → Trovo un automa o un'espressione regolare

Il linguaggio $\{a^n b^n : n \geq 0\}$ è regolare?

Risposta: NO

7)

Ordina le mosse del gioco del Pumping Lemma

1) Giocatore 1 sceglie una lunghezza
2) Giocatore 2 sceglie la parola
3) Giocatore 1 spezza la parola
4) Giocatore 2 sceglie un esponente

Il linguaggio $L = \{a^\ell b^m c^n \mid \ell, m, n \geq 0$ e se $\ell = 1$ allora $m = n\}$ è regolare?

Risposta: SI

Dim: Supponiamo L sia non regolare
1) Sia k la lunghezza del P.L.
2) Consideriamo $w=a^l b^k c^k$ $\quad$ $l+m < k$
3) Siano x, y, z tale che $w=xyz$, $y \neq \varepsilon$, $|xy|<=k$
4) Se y contiene almeno una b → $i=z$ $xy^2z \notin L$

$x=\varepsilon$, $\quad y=a$, $\quad z=b^k c^k$
Quando l'esponente è diverso da 1, allora è regolare
Di fatto rispetta le condizione del PL, ma non è regolare
$xyz= a^i b^k c^k \in L$

Dim: L rispetta le condizioni del P.L.
1) La lunghezza k=1
2) sia $w=a^l b^m c^n \in L$
3) considero che $x=\varepsilon$, $y=a$, $\quad$ se $l > 0$ $z=a^{l-1}b^m c^n$

se $l=0$ $\quad$ $y=b$ e $m > 0$
se $m=0$ $\quad$ $y=c$

4) $\forall$ i, $xy^iz \in L$
   se $y=b$ → $xy^iz = b^i b^{m-1} c^n \in L$
   se $y=\varepsilon$ → $ac^l c^{n-1} b \in L$

   $y=a$ se $l=1$ → $xy^iz= a^l b^m c^n \in L$
   se l diverso da 1
   $xy^l= a^i a^{l-1} b^n c^m \in L$

8)

Qual è il linguaggio generato dalla grammatica G_1?

Risposta:

$0^n\#1^n \mid n \geq 0$

Associa la notazione con la definizione:

$A \to uAv$

B. Regola

$uAv \Rightarrow uwv$

A. Produzione

$S \Rightarrow^* u$

C. Derivazione

Descrivi il linguaggio generato dalla grammatica $S \to (S) \mid SS \mid \varepsilon$

Si intende:
S → (S)
S → SS
S → ε
La regola dice che per ogni parentesi aperta ci sta una parentesi chiusa:
S → (S) → ((S))
→ (((S))) → ((()))
S → SS → (S)S → (SS)S → ((S)S)S) → (()S)S → (()(S))S →
(()())S → (()())(S) → (()())((S)) → (()())(())

Descrivi il linguaggio generato dalla grammatica

$$\langle EXPR \rangle \to \langle EXPR \rangle + \langle TERM \rangle \mid \langle TERM \rangle$$
$$\langle TERM \rangle \to \langle TERM \rangle \times \langle FACTOR \rangle \mid \langle FACTOR \rangle$$
$$\langle FACTOR \rangle \to (\langle EXPR \rangle) \mid a$$

<EXPR> → <EXPR> + <TERM>
<EXPR> → <TERM>
<TERM> → <TERM> X <FACTOR>
<TERM> → <FACTOR>
<FACTOR> → (<EXPR>)
<FACTOR>→ a
Risposta: Insieme di somme e moltiplicazioni
a + a x a
(a + a) x a
(a + a + a) x (a x a) + (a+a)n

9)

Quali parole associ al termine "pila"?

Attenzione, la votazione è chiusa

Stack

Una pila è una memoria di tipo:

La risposta corretta era

LIFO (Last In, First Out)

FILO (First In, Last Out)

Quali tipi di simboli si possono trovare nella pila di un PDA?

1) Terminali 2) Variabili 3) Sia (1) che (2)

La risposta giusta è:

Nessuna delle precedenti

L'automa a pila si distingue per: *la presenza di una pila*

L'INPUT della funzione di transizione è costituito da (puoi scegliere più di una risposta):

lo stato attuale

il simbolo di input

il simbolo in cima alla pila

L'OUTPUT della funzione di transizione è:

un'insieme di coppie stato/simbolo della pila

10)

Considera la grammatica G per le espressioni aritmetiche:

$$E \to E + T \mid T$$
$$T \to T \times F \mid F$$
$$F \to (E) \mid a$$

Costruisci il PDA equivalente alla grammatica.

Quante sono le transizioni *estese* che vanno da q_{loop} a q_{loop}?

ε, ε → T

ε, ε → E + T

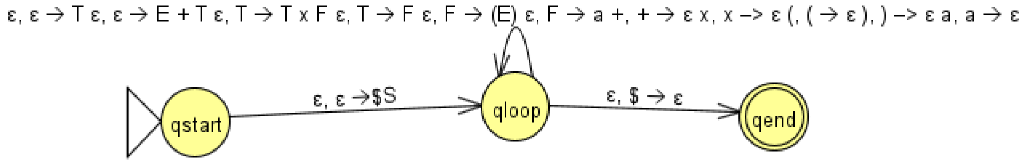
ε, T → T x F

ε, T → F

ε, F → (E)

ε, F → a
considerando i 5 simboli terminali
+, + → ε
x, x –> ε
(, (→ ε
),) –> ε
a, a → ε
avremo 6+5=11 simboli.

Quindi se rappresentato sarebbe:



Come va modificato il PDA per poterlo trasformare in grammatica?

Risposta: Deve svuotare la pila prima di accettare

La variabile A_{pq} genera tutte le stringhe x tali che

Risposta: x porta il PDA da “p” con pila vuota a “q” con pila vuota

La regola $A_{pq} \to aA_{rs}b$ corrisponde al caso

Risposta: Il simbolo inserito nella pila all’inizio viene rimosso alla fine di *x*.

La regola $A_{pq} \to A_{pr}A_{rq}$ corrisponde al caso

Risposta: Il simbolo inserito nella pila all’inizio viene rimosso prima della fine di *x*.

11)

Quali dei seguenti linguaggi sono context-free? Puoi scegliere più di una risposta

- $\{a^n b^n \mid n \geq 0\}$
- $\{a^n b^n c^n \mid n \geq 0\}$
- $\{a^n b^j a^n b^j \mid n, j \geq 0\}$
- $\{a^n b^j a^j b^n \mid n, j \geq 0\}$
- $\{ww \mid w \in \{0,1\}^*\}$
- $\{ww^R \mid w \in \{0,1\}^*\}$

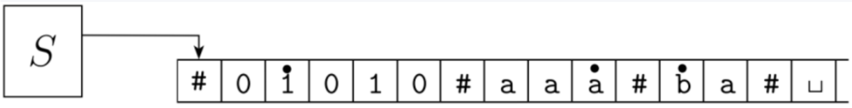
1) Si
2) No
3) No
4) No
5) No
6) Si

12)

ESPERIMENTO CONCETTUALE:
Se cambiassimo la funzione di transizione di una TM in $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R, S\}$, dove S significa "stai fermo", il nostro modello sarebbe più potente?

Risposta: No

Se nella macchina multinastro $\delta(q, 1, a, b) = (p, 0, a, a, L, R, R)$, quale sarà il contenuto del primo nastro virtuale dopo la simulazione della transizione?



Risposta:

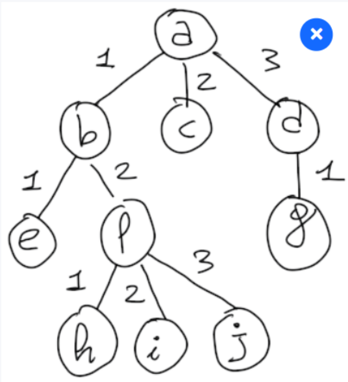
$\#\dot{0}0010\#$

Cosa succede sul secondo nastro virtuale?
Risposta

$\#aaa\dot{\_}\#$

Se nella macchina multinastro $\delta(q, 1, a, b) = (p, 0, a, a, L, R, R)$, quale sarà il contenuto del terzo nastro virtuale dopo la simulazione della transizione?

Risposta: $\#a\dot{a}\#$



Dato l'albero in figura, associare ogni stringa numerica al nodo corrispondente nell'albero

ε – a
123 – i
12 – f
111-non esiste
3 - d

La TM deterministica che simula una TM non deterministica visita l'albero di computazione:
Risposta: In ampiezza

Quale dei nastri della TM deterministica D che simula una TM non deterministica N non viene mai scritto:
Risposta: Il primo, che tiene una copia nell'input

Cosa fa la TM deterministica D che simula una TM non deterministica N dopo aver sostituito la stringa sul nastro 3 con la stringa successiva rispetto alla vista in ampiezza dell'albero?
Risposta: Ricopia il nastro 1 sul nastro 2 e ricomincia la simulazione di N dall'inizio della computazione

In quale caso la TM deterministica D che simula una TM non deterministica N termina la computazione accettando l'input?
Risposta: appena si incontra una configurazione di rifiuto per N

In quale caso la TM deterministica D che simula una TM non deterministica N termina la computazione rifiutando l'input?
Risposta: se l'albero di computazione di N è infinito e non contiene configurazioni di accettazione

In quale caso la TM deterministica D che simula una TM non deterministica N non termina la computazione?
Risposte:

se tutti i rami dell'albero di computazione di N sono infiniti

se l'albero di computazione di N è infinito e non contiene configurazioni di accettazione

Una TM con "resta ferma" invece di "muovi a sinistra" ha una funzione di transizione:

$$\delta : Q \times \Gamma \mapsto Q \times \Gamma\{S, R\}$$

Quale classi di linguaggi riconosce?

Risposta:
I linguaggi regolari

13)

Problemi di Hilbert: https://it.wikipedia.org/wiki/Problemi_di_Hilbert

La TM decide o riconosce il linguaggio $D = \{p \mid p$ è un polinomio su x con radice intera$\}$?

$M_1 =$ **"Su input $\langle p \rangle$, polinomio sulla variabile x:**

1. Valuta p con x posta successivamente ai valori 0, 1, -1, 2, -2, ... Se in qualche momento la valutazione del polinomio è 0, accetta.

Risposta: Riconosce

Cosa c'è che non va in questa macchina di Turing?

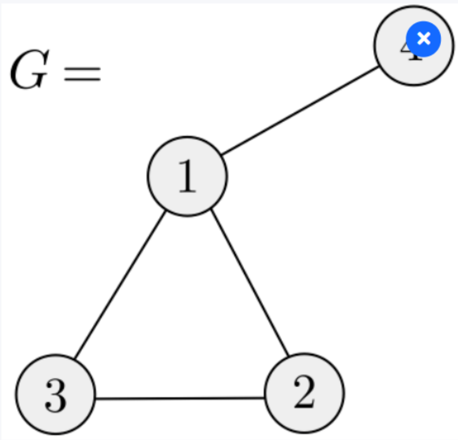
$M_{bad} =$ **"Su input $\langle p \rangle$, polinomio sulle variabili $x_1, \ldots, x_k$:**

1. Per tutte possibili assegnazioni con valori interi di $x_1, \ldots, x_k$

2. Valuta p per ogni assegnazione

3. Se una di tali assegnazioni dà valore 0, accetta; altrimenti, rifiuta."

Risposta:
Questa macchina va sempre in loop in quanto continuerà a valutare il polinomio su tutte le possibili infinite combinazioni di input, memorizzandosi che ha trovato un certo valore e rimanendo lì finché non trova ciò che cerca. Questa TM è solo un riconoscitore, in quanto per come è scritta non accetterà mai.



Scrivi la codifica $\langle G \rangle$ del grafo G mostrato in figura.

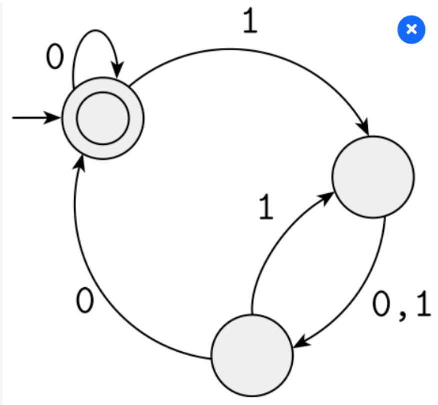
- (1,2,3,4)((1,2),(1,3),(1,4),(2,3))

Il grafo H con codifica $\langle H \rangle = (1,2,3,4,5)((1,2),(2,3),(3,1)(4,5))$ è connesso?

Risposta: NO

14)

<u>Wooclap</u>



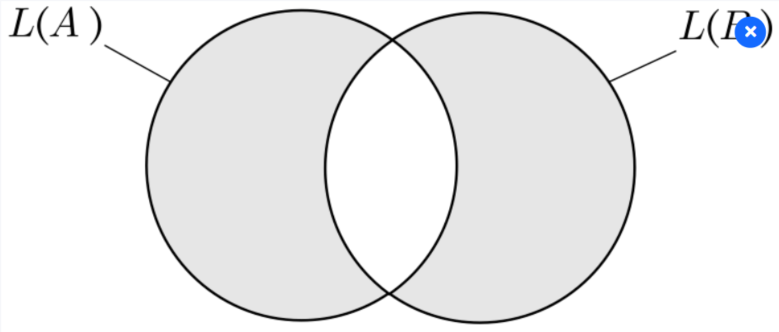
Dato il DFA M in figura, indica quali delle seguenti stringhe appartengono al linguaggio A_{DFA} (puoi scegliere più di una risposta)

Risposte: <M , 0100> <M, 000>

Questa TM è un riconoscitore o un decisore per E_{DFA}?
$T' =$ "Su input $\langle A \rangle$, dove A è un DFA:
1. Per ogni stringa $w \in \Sigma^*$, seguendo l'ordine lessicografico:
2. Esegue la TM M che decide A_{DFA} su $\langle B, w \rangle$. Se M accetta, RIFIUTA
3. Se nessuna stringa viene accettata, ACCETTA"

Risposta: Essendo una macchina che va in loop quando dovrebbe accettare e accetta quando dovrebbe fermarsi. Dunque non può essere certo un decisore, ma neanche un riconoscitore.
Riconosce quindi un linguaggio diverso.

E quindi: nessuna delle precedenti



Nella figura, la parte in grigio è la *differenza simmetrica* di $L(A)$ e $L(B)$. Cosa succede alla differenza simmetrica quando $L(A) = L(B)$?

Risposta: L'insieme vuoto - Ø

Considera i linguaggi context-free $A = \{a^n b^n c^m \mid m, n \geq 0\}$ e $A = \{a^m b^n c^n \mid m, n \geq 0\}$. Quale linguaggio ottieni se fai l'INTERSEZIONE di A e B?

Risposta: $a^n b^n c^n$, n >= 0

Sia $A_{\varepsilon CFG} = \{\langle G \rangle \mid G$ è una CFG che genera $\varepsilon\}$.
Completa la descrizione della seguente TM che decide $A_{\varepsilon CFG}$:

Risposta:
V = Su input <G>, dove G è una CFG:
1) Esegue la TM che decide A_CFG su input <G, ε>
2) Se questa macchina accetta, *accetta* altrimenti *rifiuta*

15)

Considera i seguenti insiemi: quale dei due è più grande?
$\mathbb{N} = \{0, 1, 2, \ldots, \}$, **insieme dei numeri naturali**
$\mathbb{E} = \{0, 2, 4, \ldots, \}$, **insieme dei numeri pari**

Risposta: Hanno la stessa dimensione

L'insieme $\mathbb{Q} = \{\frac{m}{n} \mid m, n \in \mathbb{N}\}$ dei numeri razionali positivi è numerabile?

Risposta: Non è numerabile

L'insieme $\mathbb{R}$ dei numeri reali è numerabile?

Risposta: Non è numerabile

Dato un alfabeto finito Σ, Σ^* è numerabile?

Risposta: Non è numerabile

L'insieme di tutte le macchine di Turing è numerabile?

Risposta: É numerabile

L'insieme di tutte le sequenze binarie infinite è numerabile?

Risposta: Non è numerabile

Dato un alfabeto finito Σ, l'insieme di tutti i linguaggi su Σ^* è numerabile?

Risposta: Non è numerabile

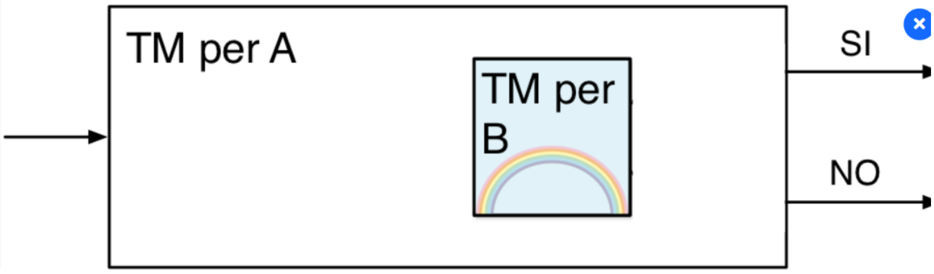
16)

Dato n, questa funzione termina?

```
void collatz(int n)
{
while(n > 1)
{
if(n % 2 == 1)
n = 3*n + 1;
else
n = n / 2;
}
return;
}
```

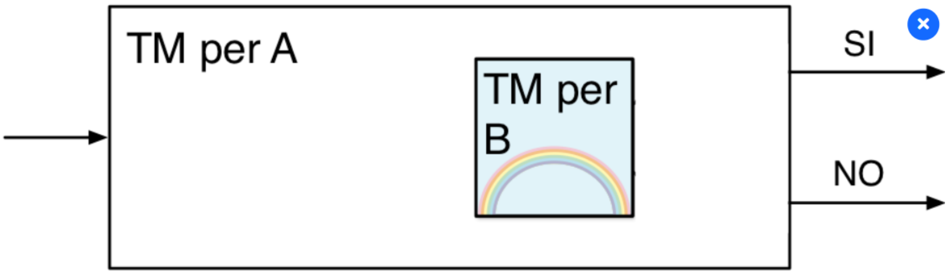
Risposta: Deriva da una congettura del matematico Collatz.
Maggiori al link: https://it.wikipedia.org/wiki/Congettura_di_Collatz
Risposta giusta quindi: *Non so*
Di fatto è un *halting problem* e si dimostra che in determinate condizioni, il calcolo di questa particolare successione può effettivamente fermarsi. D'altro canto, essa è un ciclo infinito descritto da una funzione e da un albero caratteristico.

Supponiamo che A sia riducibile a B: questo vuol dire che possiamo usare una soluzione per B per risolvere il problema A. Se B è *DECIDIBILE*, allora:



Risposta: A è decidibile

Supponiamo che A sia riducibile a B: questo vuol dire che possiamo usare una soluzione per B per risolvere il problema A. Se B è *INDECIDIBILE*, allora:



Risposta: A è indecidibile

Supponiamo che A sia riducibile a B: questo vuol dire che possiamo usare una soluzione per B per risolvere il problema A. Se A è *DECIDIBILE*, allora:

Risposta: Non possiamo dire nulla su B

Supponiamo che A sia riducibile a B: questo vuol dire che possiamo usare una soluzione per B per risolvere il problema A. Se A è *INDECIDIBILE*, allora:

Risposta: B è indecidibile

Sappiamo che A_{TM} è indecidibile. Possiamo usare questo fatto per dimostrare che anche $HALT_{TM}$ è indecidibile?

Risposta:

Si, costruendo una TM che risolve A_{TM} usando una TM per $HALT_{TM}$ come subroutine

Data una TM M e una parola w, costruiamo la seguente TM M_1:
$M_1 =$ "Su input x:
1. Se $x \neq w$, RIFIUTA
2. Se $x = w$, esegui M su w e ACCETTA se M accetta. Altrimenti RIFIUTA."
Quale delle seguenti affermazioni è vera?

Risposta:
$L(M_1) = \{w\}$ se M accetta w
$L(M_1) = \emptyset$ se M non accetta w

Data una TM M e una parola w, costruiamo la seguente TM M_2:
$M_2 =$ "Su input x:
1. Se x ha la forma 0^n1^n, ACCETTA
2. Se x non ha tale forma, esegui M su w e ACCETTA se M accetta w. Altrimenti RIFIUTA."
Quali delle seguenti affermazioni sono vere? (puoi scegliere più di una risposta)

Risposte:

$L(M_2) = \{0,1\}^*$ se M accetta W
$L(M_2) = 0^n1^n$ se M non accetta W

Completa la descrizione della TM S che decide E_{TM} usando la TM R che decide EQ_{TM}.
S deve accettare $\langle M \rangle$ se $L(M) = \emptyset$, e rifiutare quando $L(M) \neq \emptyset$

$S =$ "Su input $\langle M \rangle$, dove M è una TM:
1. ......

17)

Supponiamo che A sia riducibile mediante funzione a B. Questo vuol dire che esiste una funzione calcolabile che trasforma istanze di A in istanze di B. Quali delle seguenti affermazioni sono vere?

Risposte:

Se B è decidibile, allora A è decidibile

Se A è indecidibile, allora B è indecidibile

Per dimostrare che $A_{TM} \leq_m HALT_{TM}$ dobbiamo trovare una funzione calcolabile f che rispetta certe proprietà. Qual è l'INPUT della funzione di riduzione?

Risposta: Una TM e una stringa

<M, w>

Sarebbe l'OUTPUT nella domanda:

Per dimostrare che $A_{TM} \leq_m HALT_{TM}$ dobbiamo trovare una funzione calcolabile f che rispetta certe proprietà. Qual è l'INPUT della funzione di riduzione?

Risposta: Una TM e una stringa <M, w>

Per dimostrare che $A_{TM} \leq_m E_{TM}$ dobbiamo trovare una funzione calcolabile f che rispetta certe proprietà. Qual è l'OUTPUT della funzione di riduzione?

Risposta: Una TM e una stringa <M, w>

Per dimostrare che $A_{TM} \leq_m E_{TM}$ dobbiamo trovare una funzione calcolabile f che rispetta certe proprietà. Qual è l'INPUT della funzione di riduzione?

Risposta: Una TM <M'>

Per dimostrare che $A_{TM} \leq_m E_{TM}$ dobbiamo trovare una funzione calcolabile f che rispetta certe proprietà. Qual è la relazione tra input e output della funzione di riduzione?

Risposta: M accetta w se e solo se L(M') = Ø

Supponiamo che A sia riducibile mediante funzione a B. Questo vuol dire che esiste una funzione calcolabile che trasforma istanze di A in istanze di B. Quali delle seguenti affermazioni sono vere?

Risposte:

Se A non è Turing-riconoscibile, allora B non è Turing-riconoscibile

Se B è Turing-riconoscibile, allora A è Turing-riconoscibile

Quale riduzione possiamo usare per dimostrare che EQ_{TM} **non** è Turing-riconoscibile?

Risposta:

$$\overline{A_{TM}} \leq_m EQ_{TM}$$

Quale riduzione possiamo usare per dimostrare che EQ_{TM} **non** è coTuring-riconoscibile?

Risposta:

$A_{TM} \leq_m EQ_{TM}$

18)

Considera il linguaggio $A = \{0^n 1^n \mid n \geq 0\}$. Quanto tempo serve ad una Macchina di Turing per decidere il linguaggio?

Indica quali delle seguenti affermazioni sono vere. Puoi scegliere più di una risposta.

La risposta corretta era

Per la terza risposta

With some algebra (and changing the constant in the $O(n)$), we can

$2n = O($

$n \log n =$

$3^n = 2^{O(}$

Rispondete alla domanda di Zio Paperone: quanti chicchi di riso ci sono sulla scacchiera alla fine dei raddoppi?

Risposta:

$2^{64} - 1$

Questa TM decide PATH in tempo polinomiale?

P = "su input $\langle G, s, t \rangle$, dove G è un grafo, s, t, sono vertici di G:
1. Considera tutti i cammini che iniziano da s di lunghezza crescente $1, 2, \ldots, n$, dove n è il numero di vertici del grafo.
2. Se uno dei cammini termina in t, ACCETTA
3. Se nessuno dei cammini termina in t, RIFIUTA."

Partendo da n
1 * (n – 1) * (n – 2) * (n – 3) ……… 2 * 1
È un algoritmo più che polinomiale

19)

Domino[1] è un problema che appartiene a P?

Risposta: Si

Domino[2] è un problema che appartiene a P?

Risposta: No

20)

Associa ogni classe di complessità alla sua defìnzione

P
A. linguaggi che si possono decidere in tempo polinomiale

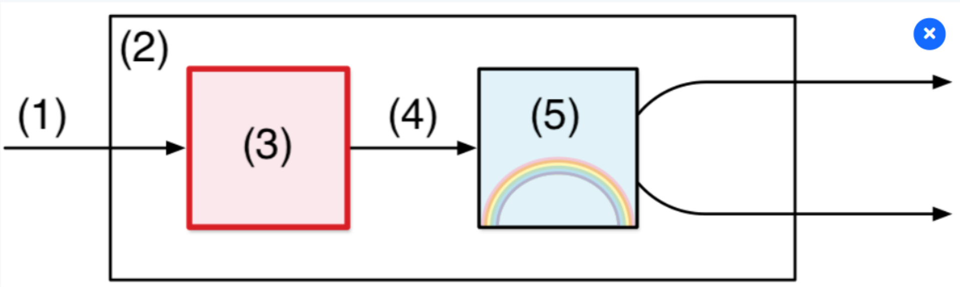
NP
B. linguaggi che si possono verificare in tempo polinomiale

L'esistenza di un algoritmo polinomiale per un problema NP-Hard implica l'esistenza di un algoritmo polinomiale per ogni problema di complessità:

Risposta: NP

Un problema NP-completo è un problema che è contemporaneamente ….. e ……
Seleziona due tra le scelte possibili:

Risposte:
- in NP
- in NP-Hard



Completa lo schema di riduzione che dimostra che B è un problema NP-Hard usando A come problema NP-Hard di riferimento. Associa ogni etichetta alla sua posizione nella figura.

Risposte:

1) Istanza di A
2) TM per A
3) Riduzione polinomiale
4) Istanza di B
5) TM per B

Disponi i passi della dimostrazione che SAT è NP-Hard:

1) Scelgo CircuitSAT come problema che so essere NP-Hard
2) Descrivo una funzione di riduzione polinomiale da CircuitSAT a SAT
3) Dimostro che la funzione di riduzione è corretta
4) Dimostro che la funzione di riduzione impiega tempo polinomiale
5) Siccome CircuitSAT ≤P SAT e CircuitSAT è NP-Hard, allora SAT è NP-Hard