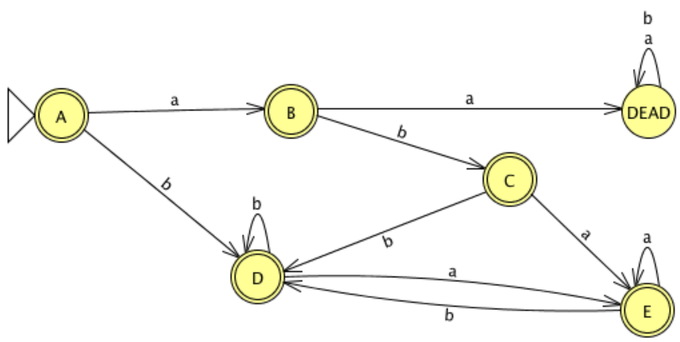
# Teoria degli Automi e Calcolabilità a.a. 2022/23
## Prova scritta 12 settembre 2023

**Esercizio 1** Minimizzare il seguente DFA, descrivendo in modo preciso i passaggi effettuati:




**Soluzione** Inizialmente si hanno le due classi di equivalenza $DEAD$ (non finale) e tutti gli altri stati (finali). Possiamo distinguere $B$ dagli altri stati finali perché leggendo $a$ è l'unico che va in uno stato non finale, quindi otteniamo le classi di equivalenza $\{DEAD\}$, $\{B\}$ e $\{A, C, D, E\}$. Ora possiamo distinguere $A$ da $C, D, E$ perché leggendo $a$ si va in $\{B\}$, mentre per gli altri tre stati si va in $\{A, C, D, E\}$. A questo punto non possiamo distinguere ulteriormente perché anche leggendo $b$ da $C, D, E$ si resta in $\{C, D, E\}$. L'automa minimo è quindi:

| | $a$ | $b$ |
|---|---|---|
| $\rightarrow *\{A\}$ | $\{B\}$ | $\{C, D, E\}$ |
| $*\{B\}$ | $\{DEAD\}$ | $\{C, D, E\}$ |
| $*\{C, D, E\}$ | $\{C, D, E\}$ | $\{C, D, E\}$ |
| $\{DEAD\}$ | $\{DEAD\}$ | $\{DEAD\}$ |

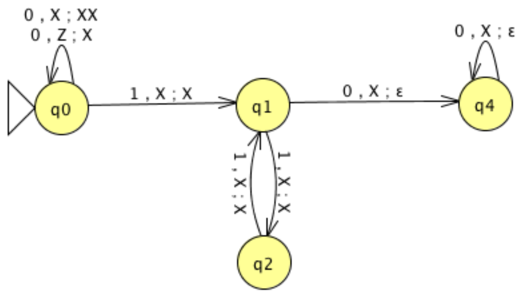
**Esercizio 2** Provare che il linguaggio $\{0^n 1^k 0^n \mid k \text{ dispari}, n > 0\}$ non è regolare.

**Soluzione** Possiamo dimostrarlo utilizzando il pumping lemma. Infatti, preso $n > 0$ arbitrario[1], consideriamo la stringa $0^n 1 0^n$ che appartiene al linguaggio ed è di lunghezza $\geq n$. Decomponenendo questa stringa come $uvw$ con $|uv| \leq n$ e $v \neq \epsilon$, si ha che sicuramente le stringhe $u$ e $v$ contengono solo 0. Allora la stringa $uv^0 w$ contiene un numero di 0 strettamente minore di quello in $uvw$, quindi non appartiene al linguaggio.

**Esercizio 3** Dare un automa a pila che riconosca (per pila vuota) il linguaggio dell'esercizio precedente. È possibile dare un automa deterministico?

**Soluzione** Una soluzione è la seguente:

[1] Per $n = 0$ possiamo prendere la stringa 010.




Questo automa è deterministico.

**Esercizio 4** Si classifichino i seguenti problemi (ricorsivo, ricorsivamente enumerabile ma non ricorsivo, non ricorsivamente enumerabile), giustificando le risposte.

1. Determinare se un programma restituisce (per qualche input) uno 0 in output.
   Questo problema ($\{x \mid \phi_x(y) = 0$ per qualche $y\}$) non è ricorsivo per il teorema di Rice (estensionale e non banale). È ricorsivamente enumerabile in quanto è possibile eseguire il programma per tutti i possibili input con la tecnica a zig-zag e se per qualche input il programma restituisce 0 questo sarà trovato.

2. Determinare se un programma non restituisce mai uno 0 in output.
   Analgamente al caso precedente questo problema ($\{x \mid \phi_x(y) > 0$ per ogni $y\}$) non è ricorsivo per il teorema di Rice (estensionale e non banale). Non è ricorsivamente enumerabile in quanto è il complementare del precedente che lo è.

3. Determinare se un programma (assumiamo sia una macchina di Turing) ha solo mosse che spostano a sinistra.
   Questo problema è ricorsivo in quanto è possibile dare un algoritmo che esamina (il codice = tabella di transizione di) una macchina e controlla se esistono mosse che spostano a sinistra.

**Esercizio 5** Siano $\mathcal{P}$ e $\mathcal{Q}$ due insiemi ricorsivamente enumerabili, e indichiamo con $\mathcal{A}^k_{\mathcal{P}}$ l'esecuzione di $k$ passi dell'algoritmo che semidecide $\mathcal{P}$, e analogamente per $\mathcal{Q}$. Si descriva in pseudocodice un algoritmo che semidecide $\mathcal{P} \cup \mathcal{Q}$.

```
input x
k = 0
while(true)
  if (A^k_P(x)==1) || A^k_Q(x)==1) return 1
  k++
```