Sia A un linguaggio, e sia $DROPOUT(A)$ come il linguaggio contenente tutte le stringhe che possono essere ottenute togliendo un simbolo da una stringa di A:

$$DROPOUT(A) = \{xz \mid xyz \in A \text{ dove } x, z \in \Sigma^* \text{ e } y \in \Sigma\}.$$

Mostrare che la classe dei linguaggi regolari è chiusa rispetto all'operazione $DROPOUT$, cioè che se A è un linguaggio regolare allora $DROPOUT(A)$ è un linguaggio regolare.

Per dimostrare che la classe dei linguaggi regolari è chiusa rispetto all'operazione DROP OUT, dobbiamo provare che, se A è un linguaggio regolare, allora DROP OUT(A) è anch'esso un linguaggio regolare.

Possiamo procedere nel seguente modo:

1. Supponiamo che A sia un linguaggio regolare. Ciò significa che esiste un automa a stati finiti NFA/DFA che riconosce A.

2. Costruiamo un nuovo automa a stati finiti che riconosca $DROPOUT(A)$ a partire dall'automa che riconosce A.

3. L'idea è di creare un nuovo automa che simuli l'esecuzione dell'automa originale per A, ma che accetti una stringa se e solo se è possibile saltare (drop out) un simbolo in qualche punto della computazione.

4. Il nuovo automa avrà un insieme di stati che è il prodotto cartesiano dell'insieme di stati dell'automa originale con l'alfabeto Σ. Ogni stato (q, a) del nuovo automa rappresenta lo stato q dell'automa originale dopo aver letto il simbolo a.

5. Le transizioni del nuovo automa sono definite come segue:

- Per ogni transizione $\delta(q, b) = p$ nell'automa originale, il nuovo automa avrà una transizione $\delta'((q, a), b) = (p, b)$ per ogni a in Σ.
- Inoltre, per ogni transizione $\delta(q, b) = p$ nell'automa originale, il nuovo automa avrà una transizione $\delta'((q, a), \varepsilon) = (p, b)$ per ogni a in Σ. Questa transizione rappresenta il fatto che il simbolo b è stato "saltato" (dropped out).

6. Lo stato iniziale del nuovo automa sarà (q_0, ε), dove q_0 è lo stato iniziale dell'automa originale e ε rappresenta la stringa vuota.

7. Gli stati finali del nuovo automa saranno tutti gli stati della forma (q, a) dove q è uno stato finale dell'automa originale e a è un simbolo qualsiasi dell'alfabeto Σ.

Questo nuovo automa riconosce esattamente il linguaggio $DROPOUT(A)$. Infatti, accetta una stringa xz se e solo se esiste una stringa xyz in A tale che y è un singolo simbolo.

Poiché l'automa è stato costruito a partire da un automa che riconosce un linguaggio regolare A, e le operazioni eseguite per costruirlo preservano la regolarità, il nuovo automa è anch'esso un automa a stati finiti. Quindi, $DROPOUT(A)$ è un linguaggio regolare.

2. **(8 punti)** Per ogni linguaggio L, sia substring$(L) = \{v \mid uvw \in L$ per qualche coppia di stringhe $u, w\}$. Dimostra che se L è un linguaggio context-free, allora anche substring(L) è un linguaggio context-free.

Innanzitutto, ricordiamo che la classe dei linguaggi liberi da contesto è chiusa rispetto alle operazioni di unione, concatenazione e chiusura di Kleene. Questo significa che, se L1 e L2 sono CFL, allora anche L1 ∪ L2, L1.L2 e L1* sono CFL.

Ora, consideriamo il linguaggio $substring(L) = \{v \mid uvw \in L$ per qualche coppia di stringhe $u, w\}$.

Possiamo costruire un linguaggio che genera substring(L) nel seguente modo:

1. Creiamo una nuova variabile iniziale S' che non appartiene alla grammatica G che genera L.
2. Aggiungiamo le seguenti nuove produzioni a G: $S' \rightarrow ASB \mid BSA \mid S$ dove $A \rightarrow \varepsilon \mid aA$ (a è un simbolo terminale dell'alfabeto) e $B \rightarrow \varepsilon \mid bB$ (b è un simbolo terminale dell'alfabeto)
3. Le produzioni di S' generano tutte le possibili stringhe della forma uSv, dove u e v sono stringhe arbitrarie (generate da A e B rispettivamente) e S è la variabile iniziale della grammatica G che genera L.
4. Quindi, il linguaggio generato dalla nuova grammatica con variabile iniziale S' è l'insieme di tutte le stringhe della forma uvw, dove v è una sottostringa di una stringa w appartenente a L.

Dimostriamo che questo linguaggio è effettivamente substring(L):

- Se una stringa uvw è generata dalla nuova grammatica, allora v è una sottostringa di una stringa w appartenente a L, per costruzione. Quindi, uvw appartiene a $substring(L)$.
- Viceversa, se una stringa uvw appartiene a $substring(L)$, allora esiste una stringa xyz in L tale che v è una sottostringa di xyz. Quindi, la nuova grammatica può generare uvw sostituendo v al posto di S nella derivazione di xyz.