Descrivere, anche tramite un disegno, la relazione che esiste tra i linguaggi regolari, i linguaggi CF, i linguaggi CF non inerentemente ambigui, i linguaggi riconosciuti dai PDA che accettano per stato finale, quelli riconosciuti dai PDA che riconoscono per pila vuota e, per finire, quelli riconosciuti dai DPDA che accettano per stato finale e quelli riconosciuti dai DPDA che accettano per pila vuota. La risposta va motivata in modo conciso.

Tutte queste caratterizzazioni sono composte da linguaggi che accettano un insieme comune di operazioni chiuse, descrivendo principalmente unione, concatenazione, chiusura di Kleene, ecc.
Le varie operazioni sono definite induttivamente, tali che:

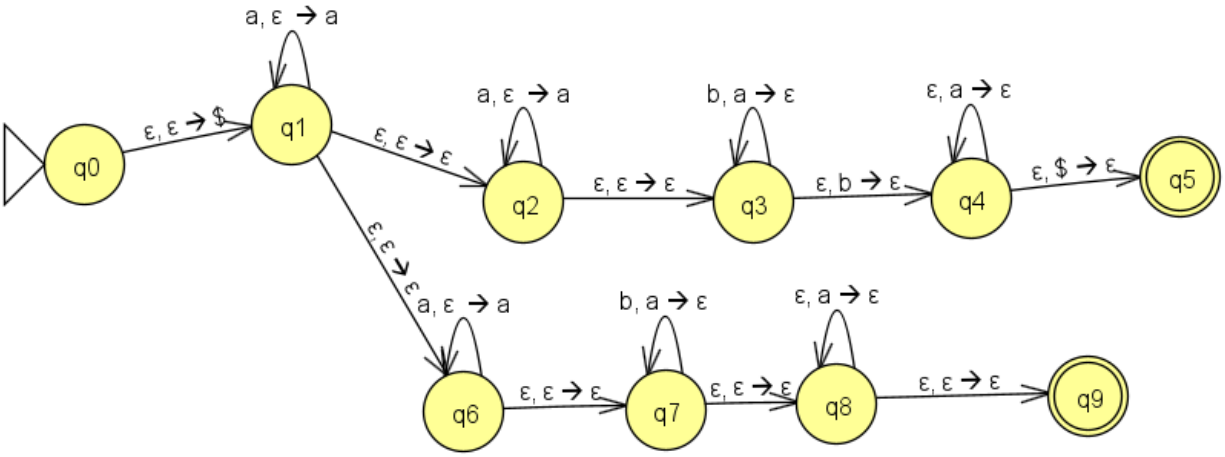
- nei linguaggi regolari danno luogo ad un automa, tale che esista almeno un ciclo (PL)
- nei linguaggi CF, analogamente, esiste una sottoderivazione ricorsiva (PL per i CF)
- nei linguaggi per PDA, essi avranno necessariamento uno stato iniziale, svuotato alla fine della pila ed un insieme di stati finali, tali che le transizioni regolari siano composte da push/pop degli stati tolti alla fine (pila vuota).

Tutti quindi presentano stati ripetuti, sceglibili deterministicamente/indeterministicamente, tali da poter avere ripetizione di stringhe e scelta possibile a seconda delle proprietà del linguaggio.

2. Descrivere un PDA, che accetta per pila vuota, che riconosca il seguente linguaggio $L = \{a^n b^m \mid 0 \leq n \leq m \leq 2n\}$. E' possibile costruire il PDA passando prima per una CFG che genera L. In questo caso è richiesta una dimostrazione o almeno una spiegazione convincente del fatto che la CFG generi veramente L.

Il linguaggio è costituito da un numero diverso di “a” e “b”, avendo in particolare un numero doppio come massimale di “a” rispetto a quello delle “b”.
L’ordine di queste stringhe deve essere mantenuto, avendo idealmente un numero di “a” uguale a quelle delle “b” facendo pop di un singolo per volta, oppure eseguendo il pop di due “b” e ottenendo m=2n oppure alternando le cose, avendo quindi un numero doppio di “a” rispetto alle “b” nelle normali condizioni.
Partiamo dalla descrizione di una grammatica che descrive questo linguaggio:
S → aA | ε
A → aAbB | ε
B → aB | ε
In questo modo posso avere già almeno un numero doppio di “a” rispetto alle “b” e la CFG genera effettivamente questo linguaggio, avendo ad esempio come derivazione:
S → aA → aAbB → aAbaB → aAbBaB → aab

Segue il possibile PDA di questa grammatica:
(seguendo l’idea numero di “a”=numero di “b” oppure per l’idea di prima numero doppio di “a” rispetto al numero di “b”)



Si chiede di descrivere la Forma Normale di Chomsky (FNC) e di descrivere l'enunciato del pumping Lemma, spiegando, infine, come il fatto che ogni CFG possa venire messa in FNC sia fondamentale per dimostrare il pumping Lemma.

La forma normale di Chomsky si pone l'idea di avere regole formate da derivazioni con al massimo 2 transizioni, di cui ve ne è almeno una terminale. Essendo che ogni linguaggio context-free è dimostrabile essere generato da una formale normale di Chomsky, ci si pone l'obiettivo di trasformarlo:

- aggiungendo una nuova variabile iniziale
- eliminazione delle ε-regole
- eliminazione regole unitarie A → B
- trasformazione delle regole rimaste per avere solo 2 transizioni finali

Similmente, il pumping lemma si propone, nel caso dei linguaggi CF, di avere una parola di lunghezza |w| >= k con una suddivisione w=uvxyz tali che possiamo pompare uv^ixy^iz per un i >=0
Si nota quindi che il PL sia dimostrato a causa del fatto che, essendo una grammatica ricorsiva, stanti le condizioni di prima abbiamo bisogno di almeno 2 simboli ripetuti ed 1 terminale, avremo almeno una singola ripetizione delle parole (seguendo l'idea del $B=2^V+1$ del pumping lemma, nel caso in cui almeno uno dei due simboli sia terminale, pur avendo regole vuote, da cui il +1)

4. Il Teorema di Rice tratta dell'indecidibilità delle proprietà dei linguaggi RE (cioè riconosciuti dalle Macchine di Turing). Considerate la seguente proprietà P_{REG}: il linguaggio è Regolare. Date la definizione del corrispondente linguaggio $L_{P_{REG}}$. Successivamente, spiegate se questo linguaggio è indecidibile o no. Probabilmente vi servirà il Teorema di Rice.

Assumiamo per contraddizione che P sia un linguaggio decidibile che soddisfa le proprietà e che R_P sia una TM che decide P. Mostriamo come decidere A_{TM} usando R_P costruendo una TM S.
Per prima cosa, si deve dire che P è un linguaggio non triviale; dunque, deve avere una possibile descrizione del linguaggio. Essendo regolare possiamo simularlo con un DFA tale che accetti avanzando regolarmente tutte le sue stringhe. P poi deve essere proprietà del linguaggio, dunque M_2 deve decidere le stringhe di M_1.
Essendo non triviale, si crea S che su input w:

- Usa m per costruire M_w
- M_w avanza su w. Se si ferma, significa che il linguaggio non è regolare e dunque trivialmente segnala NO come output.
- Altrimenti sfrutta R_P che in maniera non triviale cerca di decidere il linguaggio. Se accetta, regolarmente, accetta.

M_w accetta se e solo se P accetta, come proprietà del suo linguaggio.

5. "Colorare" i vertici di un grafo significa assegnare etichette, tradizionalmente chiamate "colori", ai vertici del grafo in modo tale che nessuna coppia di vertici adiacenti condivida lo stesso colore. Il problema kCOLOR è il problema di trovare una colorazione di un grafo non orientato usando k colori diversi.
   (a) Dimostrare che il problema 5COLOR (colorare un grafo con 5 colori) è in NP fornendo un certificato per il *Sì* che si può verificare in tempo polinomiale.
   (b) Mostrare come si può risolvere il problema 3COLOR (colorare un grafo con 3 colori) usando 5COLOR come sottoprocedura.

a) 5COLOR è in NP in quanto esiste un certificato in tempo polinomiale. In pratica:
- per ogni vertice controllo di avere una coppia adiacente
- per ogni coppia di vertici controlla se hanno lo stesso colore
Se accetta entrambe le condizioni, esiste un certificato in tempo polinomiale.

b) Usiamo il problema 3COLOR usando 5COLOR:
Questo è un problema risolvibile in tempo polinomiale, tale che:

- sia S un'istanza buona di 5C. Se abbiamo almeno, per il verificatore, ogni coppia di vertici conterrà un colore tra i possibili 5 esistenti, allora usando 5Color riusciamo a dire che esistono almeno 3 colori contenuti nell'insieme dei 5 possibili. Banalmente, mentre si assegnano i vertici p possibile verificare in tempo lineare che ciascuna coppia si a diversa e conterrà almeno 3 colori diversi
- sia S' un'istanza buon di 3C. Allora questi 3 colori per ciascuna coppia di vertici sono almeno contenuti in un'istanza buona di 5C. Certificando che le coppie sono tutte diverse tra di loro, risolviamo correttamente il problema.