Domanda: Descrivere il formato generico delle istruzioni in three address code e in P-code, soffermandosi sui pregi e i difetti di ciascuna delle due forme di codice intermedio.

Risposta: Il Three adress code (3AC) è uno dei modelli più usati per la generazione del codice intermedio. Il nome “Three Adress Code” deriva dal fatto che ogni istruzione contiene 3 indirizzi, due per gli operandi e uno per il risultato.
Id:=id1 operatore id2
Non tutte le istruzioni sono rappresentabili con un 3AC standard, poiché un'istruzione come l'assegnazione o la negazione richiedono solo due indirizzi. Per questo motivo non esiste una forma standardi per il 3AC.
Entrando nello specifico, un indirizzo può essere ,un nome, una costante o un indirizzo temporaneo.
Il P-COde contiene le descrizioni e le informazioni specifiche legarte alla struttura della P-machine(è costituita da una memoria per il codice, una memoria per le variabili, uno stack per le variabili temporene e una serie di registri per la gestione dello stack e il supporto dell'esecuzione)
Le istruzioni in P-Code richiedono meno indirizzi rispetto all'3AC il quale però è più compatto dell P-code in termini di numero di istruzioni.
Il P-Code non è self contained in quanto le istruzioni operano implicitamente sullo stack. Il vantaggio dell'uso dello stack sta nel fatto che il compilatore non ha bsogno di assegnare nomi ai valori che necessitano in ogni ogni punto del codice, cosa che accade in 3AC

Domanda:Descrivere un algoritmo per il calcolo degli attributi di una generica grammatica con attributi.

Risposta: le grammatiche con attributi sono grammatiche CD in cui sono state aggiunte proprietà delle entità sintattiche del linguaggio(attributi) e regole di valutazione di tali proprietà(regole semantiche).il calcolo degli attributi avviene con una semplice visita dell'albero di parsing.Il calcolo degli attributi dipende anche da come viene visitato l'albero sintattico decorato, per questo motivo introduciamo il concetto di dipendenze funzionali degli attributi. Una regola semantica assegna un valore ad un attributo di un nodo in funzione dei valori di altri attributi del nodo stesso o dei nodi vicini.
L'algoritmo che deve calcolare gli attributi deve calcolare l'attributo di un nodo prima del

calcolo dell'attributo del nodo successore(bisogna trovare un ordinamento topologico del grafo).Tale grafo deve essere aciclico.
L'ordinamento topologico del grafico consiste in una sequenza di vertici in modo tale che se esiste un arco da u a v, allora u precede v nell'ordinamento

Domanda:Si descriva l'algoritmo di partizionamento in blocchi base, specificando in quale fase della compilazione interviene e a quale scopo.

Molti generatori di codice partizionano le istruzioni in Codice intermedio in blocchi base che saranno eseguiti in sequenza. Ogni blocco base è una sequenza di istruzioni a 3 indirizzi. Il flusso di controllo può esntrare nel blocco solo attraverso la prima istruzione. I blocchi base diventano nodi di un grafo chiamato diagramma di flusso.
Nell'algoritmo di partizionamento in blocchi base si individuano le istruzioni "leader" che costituiscono l'inizio di un blocco base.
Un'istruzione leader può essere:

- La prima istruzione del codice C.I.
- Ogni istruzione di destinazione di un salto incondizionato o condizionato
- ogni istruzione immediatamente successiva a un salto .

Domanda:Discutere quali sono le principali o più usate implementazioni della tabella dei simboli e in quali fasi della compilazione intervengono la sua costruzione e la sua consultazione.

Risposta: Il compilatore durante le sue fasi deve tener traccia degli identificatori usati nel codice sorgente e dei loro attributi, queste informazioni sono memorizzate in una struttura detta Tabella dei simboli , la quale viene consultata ogni volta che si incontra un identificatore.(se esso non è presente si introduce un nuovo elemente con una serie di attributi, se invece è già presente allora se ne aggiornano evenuali suoi attributi).
La Ts è consultata in tutte le fasi del processo di compilazioni, mentre la sua costruzione avviene ,in linea teorica, a partire dall'analisi lessicale e completata nelle fasi successive.
Nell'analisi lessicale è creata una entry nella tabella per ogni nuovo identificatore incontrato , a questa entry non è associato nessun valore.

Nell'analisi sintattica e semantica la Ts viene riempita con informazioni sui tipu e sulla caratteristiche degli identificatori.
La Ts può essere implementata con un array, una lista concatenata, un albero binario di ricerca o una tabella hash.
Nell'implementazione come array i vantaggi sono la semplicità di implementazione, facile accesso alle sue componenti e lo spazio occupato in memoria.
I suoi svantaggi sono la dimensione della struttura da fissare a priori e le ricerche non molto efficienti.
Questi sono i motivi per i quali questo tipo di implementazione è raramente usato.
Un'altro tipo di implementazione è quella come lista concatenata, in questo caso la struttura è dinamica, la ricerca è lineare ma è possibile ottimizzarla.
La gestione con questa implementazione è semplica ma è inefficienete per le grosse tabelle.La TS come ABR il vantaggio che le 3 operazioni hanno un costo proporzionale all'altezza ma le frequenti cancellazioni possono sbilanciare l'albero.
La Ts come tabella hash è l'implementazione maggiormente usata dai compilatori.
Essa consiste di dimensionare la tabella in base al numero di elementi attesi ed utilizzare la funzione hash per indicizzare la tabella.
Questo tipo di organizzazione consente di implmenentare le 3 operazioni con un numero molto contenuto di accessi indipendente dalla dimensione della tabella

Domanda:Descrivere la fase di generazione del codice intermedio, specificando le caratteristiche e l’implementazione del codice a tre indirizzi.

Risposta:La generazione del codice oggetto è la fase più complessa di un compilatore perchè dipende dalle caratteristiche del linguaggio e dalle caratteristiche della target machine.La generazione del codice può prevedere una fase di ottimizzazione che può dipendere anche da particolari caratteristiche della target machine, per questo motivo prima di procedere con la generazione del codice oggetto, la maggior parte dei compilatori prevede la generazione del codice intermedio che sarà utilizzato dal back-end per la generazione del codice oggetto (questa fase fa parte del front-end del compilatore).
L'introduzione del codice intermedio consente l'indipendenza del front-end dal back-end, maggiore semplicità di traduzione nel codice oggetto e maggiore semplicità per le ottimizzazioni.Esistono diversi modelli usati per la generazione del codice intermedio, il modello più usato è quello 3AC.

Il nome 3AC deriva dal fatto che di solito ogni istruzione contiene 3 indirizzi , due per gli operandi e uno per il risultato. Tale forma è insufficiente per rappresentare tutte le caratteristiche di un linguaggio di programmazione in quenti certe istruzioni , come assegnazione e negazioni, richiedono una variante del 3AC che contiene solo due indirizzi.un indirizzo può essere un nome, una costante o un indirizzo temporaneo .Ogni istruzione può essere contrassegnata da una label simbolica.
Il 3AC non è implementato in forma testuale ma ogni istruzione in 3AC è realizzata attraverso un record con campi per gli operatori ed operandi, la sequenza delle istruzioni in 3AC è implementata atraverso un vettore o una lista concatenata.
Le istruzioni vengono implementate con record con 4 campi (1 per l'operatore e 3 per gli indirizzi degli operandi) questi record vengono chiamati quadruple.

Domanda:Descrivere la modalità generale con cui si effettua il calcolo degli attributi per le grammatiche con attributi e specificare se esistono metodi più semplici per attributi sintetizzati ed ereditati.

Risposta:Il calcolo degli attributi avviene con una semplice vista dell'albero di parsing.
Esistono classi di grammatiche con attributi non circolari:
Attributi sintetizzati ed ereditati.
Un attributo associato ad un nodo dell'albero sintattico si dice dintetizzato se il suo valore dipende solo dai valori degli attributi dei nodi figli mentre un attributo associato ad un nodo dell'albero sintattico si dice ereditato se il suo valore dipende dai valori degli attributi del nodo padre e/o dei nodi fratelli.Nel caso degli attributi sintetizzati il calcolo del valore degli attributi si ottiene con una sola visita in ordine posticipato dell'albero sintattico decorato. Invece nel caso degli attributi ereditati non è ben definito l'algoritmo di visita dell'albero in quanto bisogna ritardare l'applicazione delle regole remantiche fino al momento in cui le informazioni contestuali e il valore degli attributi lo consentono