Generatore di parser: Un generatore di parser prende in input un file che specifica la sintassi di un certo linguaggio(di solito nella forma delle regole di una grammatica CF) e produce in outup un codice che implementa il ruolo del parser.

Analisi Sintattica: In questa fase l'analizzatore sintattico (parser) usa i token per definire la struttura grammaticale della sequnza di token, ovvero determina la struttura delle singole istruzionie e controlla se sono rispettate le regole della sintassi del linguaggio.
I token diventano simboli terminali della grammatica.
Il parser adopera il codice semplificato e la symble table costruiti nella fase percedente e genera per ogni istruzione il corrispondente albero sintattico. L'output della fase di analisi sintattica è l'insieme degli alberi sintattici che descrivono il programma

Attributi mulitype: Il molti programmi è necessario avere differenti tipi per gli attributi dei token e dei somboli non terminali.
Per tae motivo si deve specificare l'intera collezione di tipi con la dichiarazione %union e poi scegliere di volta in volta il tipo più utile per i token e con i simboli non terminali.

Analisi Semantica: Usa il syntax tree e la symble table del programma sorgente per controllarne la consistenza semantica con le definizioni del linguaggio. L'analisi semantica consiste nell'interpretazione del significato delle strutture prodotte nella frase precedente controllando che esse siano legali e significative. Si controlla per esempio che le variabili coinvolte siano dichiarate e definite. Una parte importante dell'analisi semantica è il type checking dove il compilatore controlla che ogni operatore abbia gli operandi giusti. Per esempio controlla:

- controlla che l'indice di un array sia un intero
- applica le regole di visibilità degli identificatori
- applica le conversioni di tipo nel caso di operatori che possono essere applicati ad operandi di diverso tipo

produce l'albero sintattico decorato

Confronto fra semantica statica e dinamica:
La semantica statica è indipendente dai dati su cui opera il programma sorgente.
Verifica che una variabile sia dichiarata una sola volta e che venga usata coerentemente al tipo dichiarato. Rispetto delle regole che governano i tipi degli operandi nelle espressioni e negli assegnamenti. Rispetto delle regole di visibilità e univocità degli identificatori.
Correttezza delle strutture di controllo del linguaggio
rispetto dele regole di comunicazione fra i vari moduli che costituiscono il programma.
Verificare che le chiamate dei sottoprogrammi siano congruenti con le loro dichiarazioni.

La semantica dinamica è dipendente dai dati su cui opera il programma sorgente.
Controllo sui cicli infiniti e sulla derefenziazione di puntatori NULL.
Controllo sui limiti degli array e controllo sul un dato letto in input sia compatibile con il tipo della variabile a cui è destinato.

Analisi semantica staica:
L'analisi semantica statica, come quella della analisi lessicale e sintattica consta in due fasi:

- la descrizione delle analisi da eseguire
- l'implementazione delle analisi mediante opportuni algoritmi

Nell'analisi semantica a diffenrenza di quella sintattica non esiste una metodologia standard per definire o descrvere la semantica di un linguaggio.
Esiste un'enorme varietà di controlli semantici statici nei vari linguaggi.

Grammatiche con attributi: Sono grammatiche CF in cui sono state aggiunte proprietà delle entità sintattiche del linguaggio(attributi) e regole di valutazione di tali proprietà (regole semantiche).
Una grammatica con attributi specifica quindi sia azioni sintattiche che semantiche.
Le grammatiche con attributi sono utilizzabili in tutti i linguaggio di programmazione che obbediscono al principio della semantica guidata della sintassi che asserisce che la semantica non dipende dal contesto ma è strettamente legata alla sintassi.
Esse sono uno strumento molto potente per l'analisi semantica.

Attributi: un attributo è una qualunque proprietà delle entità sintattiche di un linguaggio.
Gli attributi possono variare molto rispetto al loro contenuto, alla loro complessità e principalmente in relazione al momento in cui essi sono calcolati.gli algoritmi per l'implementazione dell'analisi semantica non sono esprimibili come quelli del parsing
esempio di attributi(tipo di una variabile,valore di un'espressione,locazione di una variabile in memoria...)
gli attribui possono essere statici (calcolati al tempo della compilazione) o dinamici(calcolati al tempo di esecuzione).il calcolo di un attributo e l'associazione del suo valore ad un costrutto sintattico è detto binding dell'attributo. Il momento in cui avviene questo legame è dett obinding time. Differenti attributi possono avere binfing time diversi o anche lo stesso attributo può avere differenti binfing time che variano da linguaggio a linguaggio.

Albero sintattico decorato: Il significato delle regole semantiche per una particolare stringa può essere descritto usando l'albero sintattico associato agli attributi, il calcolo dell'apposita regola semantica è indicato all'interno del nodo.

Definizione e calcolo degli attributi: per definire un attributo basta inserire le opportune dichiarazione nella parte iniziare di qualsiasi analizzatore sintattico. Le parti destre delle regole devono essere espressioni effettivamente calcolabili al momento della derivazione della produzione. Questo vuol dire che gli attributi coinvolti devono essere già disponibili.

Il calcolo degli attributi avviene con una semplice visita dell'albero decorato. Il modo di visitare l'albero sintattico decorato può influire nella corretta valutazione degli attributi

Dipendenze funzionali degli attributi: una regola semantica assegna una vlore ad un attributo di un nodo dell'albero sintattico, in funzione dei valore di altri attributi del nodo stesso e dei nodi vicini (padre,fratelli e figli).
L'attributo dipende da altri attributi, i cui valori devono essere noti per consentire il calcolo.

Grafo delle dipendeze: data una grammatica con attributi, ad ogni regola è associato un grafo delle dipendeze. Per ogni nodo dell'albero di parsing etichettato con il simbolo X, il grago delle dipendenze avraò un nodo per ogni attributo di X. Se una regola definisce il valore dell'attributo A.b in funzione di X.c allora esisterà un arco da X.c a A.b
Algoritmo generale per il calcolo degli attributi: l'algoritmo deve calcolare l'attributo di un nodo prima del calcolo dell'attributo del nodo successore; bisogna cioè trovare un ordinamento topologico del grafo, ovvero la sequenza dei vertici in modo tale che se esiste un arco da u a v, allora u precede v nell'ordinamento, inoltre il grafo deve essere aciclico.

Classi di grammatiche: esistono classi di grammatiche con attributi non circolari:

- attributi sintetizzati

- attributi ereditati

Attribuyi sintetizzati: un attributo associato ad un nodo dell'albero sintattico si dice sintetizzato se il suo valore dipende solo dai valori degli attributi dei nodi figli, una grammatica che ha tuttiattributi sintetizzati è detta grammatica puramente sintetizzata o grammatica con S-attributi.

Attributi ereditati: 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.
Gli attributi ereditati sono utili per esprimere dipedenze di un costrutto di un linguaggio rispetto al su contesto.

Algoritmo per il calcolo degli attributi:
nel caso delle grammatiche con S.attributi il calcolo degli attributi si ottiene con una sola visista in ordine posticipando dell'albero sintattico decorato.

Nel caso di grammatiche con attributi ereditati non è chiaro l'algoritmo di visita dell'albero: infatti in questo caso bisgona ritardare l'applicazione delel regole semantiche fino al momento in cui le infomazioni contestuali e il valore degli attributi lo consentono.

Tabella dei simboli: il compilatore nelle sue varie fasi deve tener traccia degli identificatori utilizzati nel codice sorgente e dei loro attributi.
Queste informazioni sono memorizzate nella tabella dei simboli da consultare ogni volta che si incontra un identificatore. Se quest'ultimo non è presente nella tabella allora si introduce come nuovo elemento con una serie di sttributi. Se esso è già presente allora se ne aggiornano eventualmente i suoi attributi.

Quando costruire e interagire con la TS: la TS è consultata in tutte le fasi della compilazione. La costruzione della ts avviene ,in linea teorica, a partire dall'analisi lessicale e completata nelle sue fasi successive.
Nell'analisi lessicale è creata una entry nella tabella per ogni nuovo identificatore incontrato. Ad essa non è associato nessun valore.
Nell'analisi sintattica e semantica la Ts viene riempita con informazioni sui tipi e sulle caratteristiche degli identificatori
La ts viene coinvolta anche nelle fasi di generazione del codice.

Operazioni principali sulla TS:
insert per memorizzare le informazioni associate ad un identificatore
lookup per ritrovare le informazioni associate ad un determinato identificatore
delelete per rimuovere le informazioni associate ad un determinato identificatore quando esso non è più visibile.

Problemi TS: il problema principale è quello di individuare l'organizzazione più idonea della strittura dati che implementa la TS.
In genere due o piò fattori che possono influenzare la scelta dell'organizzazione di unastruttura dati è lo spazio di memoria occupato e la velocità di accesso.
Poicè la ts è consultata migliaia di volte nel corso della compilazione, la velocità di accessò avrà

priorià maggiore rispetto allo spazio di memoria occupato

Realizzazione della TS: array, lista concatenata, albero binario di ricerca, tabella hash.

TS come array: i vantaggi di tale implementazione sono la semplicità di implementazione, facile accesso alle sue componenti e lo spazio occupato in memoria relativo ai suoi dati. Mentre i suoi svantaggi sono la dimensione della struttura da fissare a priori e le ricorche non particolarmente efficienti.

TS come lista concatenata: in questo caso la struttura è dinamica.
La ricerca rimane lineare ma è possibile ottimizzarla mantenendo la lista ordinata o utilizzando il metodo della riorganizzazione dinamica(secondo questo metodo le voci a cui si accede sono spostate via via verso la testa della lista cosicchè i simboli più usati avranno un tempo di accesso inferiore) gestione semplice ma inefficiente per le grosse tabelle.

Ts come ABR: gli abr sono alberi binari tali che per ogni nodo mantengono un ordinamento ponendo nel sottoalbero di sinistra gli elementi minori e in quello di destra tutti gli elementi maggiori del nodo considetato. Con questa implementazione le tre operazioni hanno un costo proporzionale all'altezza ma le frequenti cancellazioni possono sbilanciare l'albero.

Ts ad accesso diretto:si suppone che l'universo delel chiavi associate agli n elementi da memorizzare siano interi in [0..m-1] con n<=m, assumento che le chiavi siano tutte distinte si definisce un array V di dimensione m tale che se un certo elemento x ha chiave k, allore V[k]=x
le caselle che non corrispondono ad alcuna chiave avranno NULL.
Tali chiavi sono usate come indice per spostarci nella struttura dati
vantaggi: il tempo richiesto per ogni azione è costante
svantaggi: se m è molto grande può diventare impraticabile, può comportare un enorme spreco di memoria.

Ts come tabella hash:l'idea di base è quella di dimensionare la tabella in base al numero di eleemnti attesi ed utilizzare una specia di funzione (funzione hash) per indicizzare la tabella.
Questo tipo di organizzazione consente di implementare le tre operazioni fondamentali con un numero molto contenuto di accessi indipendente dalla dimensione della tabella.
Una ricerca basata su hasking è diversa da quella basata su confronti.
Invece di spostarci nella struttura dati in funzione dell'esito dei confronti fra chiavi, cerchiamo di accedere agli alementi della tabella in modo diretto tramite operazioni aritmetiche che trasformano le chiavi in indirizzi della tabella
con l'hashing un elemento con chiave k viene memorizzato nelal cella h(k),
i vantaggi: riduciamo lo spazio necessario a memorizzare la tabella
svantaggi: perdiamo la corrispondenza tra chiavi e posizione in tabella

La funzione hash: un indirizzo hash è un possibile valore dell'indice di un vettore di elementi destinati ad ospitare gli elementi della tabella.

Type checking: si fa riferimento a due attività separate : calcolo e mantenimento delle informazioni sui tipi dei dati e controllo che ogni parte del programma abbia un senso per le regole dei tipi ammessi nel linguaggio.