**Analizzatore lessicale**

Lo scanner rappresenta un’interfaccia fra il programma sorgente e l’analizzatore sintattico o parser. Lo scanner, attraverso un esame carattere per carattere dell’ingresso, separa il programma sorgente in parti chiamate token che rappresentano i nomi delle variabili, operatori, label, ecc. il parser genera un albero sintattico le cui foglie sono i simboli terminali della grammatica, ovvero i token che lo scanner ha estratto dal programma sorgente e passato al parser. Il parser potrebbe fare direttamente anche l’analisi sintattica, ma non è conveniente in quanto la grammatica per i token è una grammatica regolare più semplice quindi di quella che tratta il parser. Lo scanner può interagire con il parser in due modi differenti:

* Lavorare in un passo separato, producendo i token in una grossa tabella in memoria di massa
* Interagire direttamente con il parser che chiama lo scanner quando è necessario il prossimo token nell’analisi sintattica

Lo scanner suddivide il programma sorgente in token. Il tipo di token è rappresentato con un numero intero unico, ed è memorizzato in una tabella. I valori delle costanti sono memorizzati in una constant table, mentre i nomi delle variabili in una symbol table. I compiti di uno scanner sono:

* Eliminare spazia bianchi, commenti ecc.;
* Isolare il prossimo token dalla sequenza di caratteri in input;
* Isolare identificatori e parole chiave
* Generare la symbol table

I token possono essere descritti in diversi modi. Spesso si utilizzano le grammatiche regolari. Un altro modo per descrivere i token è in modo ricorsivo piuttosto che generativo mediante automi a stati finiti. I linguaggi regolari vengono riconosciuti in modo efficiente attraverso automi a stati finiti. Tuttavia, sono linguaggi abbastanza limitati e con le loro grammatiche non si possono descrivere anche semplici costrutti dei linguaggi di programmazione. L’utilizzo delle grammatiche regolari è perciò limitato alla costruzione e al riconoscimento dei simboli base usati in un programma, ad esempio gli identificatori. Il riconoscimento di tali simboli è la prima fase eseguita durante la compilazione di un programma e prende il nome di analisi lessicale. Una volta terminata l’analisi lessicale sono stati individuati, ad esempio, le costanti, le variabili, etc. utilizzate nel programma e costituita quella che si chiama tabella dei simboli. L’analisi del programma prosegue poi con tecniche più complesse quali l’analisi sintattica e analisi semantica. Le grammatiche regolari possono essere usate per rappresentare scanner che sono implementati come automi a stati finiti.

Poiché realizzare un automa a stati finiti è banale sono stati progettati programmi in grado di generare automaticamente degli scanner usando un metodo formale per specificare l’automa a stati finiti.