Descrizione dei linguaggi di programmazione

La descrizione di un linguaggio avviene su quattro dimensioni:

- **Sintassi**: riguarda l'insieme di regole che permettono di costruire frasi corrette (relazione tra segni -> tra tutte le possibile sequenze di parole sull'alfabeto dato, seleziona un sottoinsieme che costituisce le frasi del liguaggio stesso);
- **Semantica**: riguarda l'attirbuzione di significato alle frasi (relazione tra segni e significato -> relazione tra frasi corrette e entità autonome che esistono indipendentemente dai segni che usiamo per descriverle). Si compone dell'insieme di tecniche che permettono di risalire da un pogramam alla funzione che il programmma stesso calcola;
- **Pragmatica**: riguarda il modo in cui le frasi corrette sono usate (relazioni tra segni, significato e utente-> frasi con lo stesso significato possono essere usate in modo diverso da utenti diversi);
- **Implementazione**: riguarda l'esecuzione di una frase corretta, rispettandone la semantica.

Sintassi

Le regole sintattiche del linguaggio specificano quali stringhe di caratteri sono legali nel linguaggio.
L'analisi del linguaggio dei lessemi avviene tramite un automa a stati finiti chiamato **riconoscitore**. Un riconoscitore è uno strumento che legge in input stringhe sull'alfabeto del linguaggio e decide se la stringa appartiene o meno al linguaggio. Il linguaggio dei lessemi viene quindi riconosciuto. L'analisi tramite riconoscitore è possibile poichè, in genenrale, il linguaggio dei lessemi è un linguaggio di tipo **regolare**.

Il linguaggio dei token (dei programmi) è in generale un linguaggio **context-free** (CF), quindi generato da una grammatica CF. É quindi possibile analizzarlo attraverso un **generatore**. Un generatore è uno strumento che genera stringhe di un linguaggio, attraverso il quale si può determinare se la sintassi di una particolare frase è sintatticamente corretta confrontandola con la struttura del generatore stesso (parser).

Attenzione: in ogni caso, i linguaggi regolari possono essere generati (grammatiche regolari) e i linguaggi CF possono essere riconosciuti (automi a pila).

Esempio: descrizione del linguaggio delle stringhe palindorme

Supponiamo di avere l'alfabeto $\Sigma = \{a, b\}$ ed il linguaggio $A = \{\sigma \in \Sigma^* | \sigma \text{ palindroma}\}$. Selezioniamo ora tutte le stringhe palindrome si Σ. La selezione può essere effettuata tramite una semplice definizione ricorsiva di stringa palindroma:

- Caso base: a, b, ε sono stringhe palindrome;
- Passo induttivo: se σ è una generica stringa palindroma, allora anche $a\sigma a$ e $b\sigma b$ sono palindorme.

In forma grammaticale, la definizione induttiva per le stringhe palindrome può essere scritta come segue:

1. $P \to \varepsilon$
2. $P \to a$
3. $P \to b$
4. $P \to a\sigma a$
5. $P \to b\sigma b$

É possibile riassumere i punti sopra come: $P \to \varepsilon|a|b|a\sigma a|b\sigma b$.

Terminologia

Lessema	parola, ovvero una stringa di caratteri su un alfabeto, con significato specifico (unità minima sintattica). Esempi: $index$, $=$, 2.
Token	frase, ovvero una sequenza (ben formata) di parole. Esempio: $index = 2$.
Programma	frase che appartiene alla categoria sintattica dei comandi.
Linguaggio	insieme di frasi.

Grammatiche context - free (libere)

Le grammatiche CF sono una tecnica fondamentale per la descrizione della sintassi dei linguaggi di programmazione.

Definizione

Una grammatica libera da contesto (CF) è una quadrupla G=<V, T, P, S> dove:

1. **V** è un insieme finito di variabili (dette anche simboli non terminali);
2. **T** è un insieme finito di simboli terminali ($V \cap T = \emptyset$);
3. **P** è un insieme finito di produzioni (o regole). Ogni produzione è della forma $A \to \alpha$, dove:
   - $A \in V$ è una variabile;
   - $\alpha \in (V \cup U)$.
4. $S \in V$ è una variabile speciale detta simbolo iniziale.

Dall'esempio precedente:

- a e b sono simboli terminali;
- σ è una variabile;
- P è un simbolo iniziale;
- Le righe 1-5 sono regole.

Nella grammatica i simboli terminali costituiscono le parole del nostro linguaggio, ovvero il vocabolario, mentre i simboli non terminali costituiscono le categorie sintattiche, e quindi i diversi tipi di elementi che possono essere usati per comporre frasi. Le produzioni rappresentano l'insieme delle regole di composizione. Il linguaggio è l'insieme di tutte le possibili frasi, ovvero le stringhe di simboli terminali (parole), generate a partire dal simbolo iniziale S, che rappresenta la categoria delle frasi legali nel linguaggio.

Una grammatica è detta libera dal contesto quando ogni regola sintattica è espressa sotto forma di derivazione di un simbolo a sinistra a partire da uno o più simboli a destra.

Notazione BNF

La BNF è un metalinguaggio utilizzato per descrivere i linguaggi di programmazione. Il suo principale impiego si ha nella descrizione delle grammatiche CF, dove:

- I simboli teminali sono raooresentati tramite parole;
- I simboli non teminali sono racchiusi in $<>$;
- Per le produzioni la $\to$ è sostituito con $::=$;

La notazione BNF è semplice ma sufficentemente potente per rappresentare i linguaggi di programmazione: rappresenta l'ordine di apparenza degli operatori, strutture annidate ad ogni livello, precedenza e associativa di operatori.

La notazione ENBF estende l'NBF permettendo di rappresentare opzioni:

- $[\,]$ indica 0 o 1 occorrenza del contenuto;
- $\{\}$ indica 0 o più occorrenze del contenuto;
- La virgoma permette di esprimere più opzioni in or.

Analisi semantica

L'analisi semantica si divide in due parti:

- Semanica statica, che determina i contesti in cui le stringhe sintatticamente corrette sono illegali (es: la stringa $I := R + 3$ è corretta, ma se il linguaggio prevede la dichiarazione del tipo di una variabile e I non fossa stata dichiarata, sarebbe un errore);
- Semantica dinamica, che si occupa di attribuire un significato [1] ad ogni frase sintatticamente corretta.

L'analisi semantica è più complessa rispetto a quella sintattica, poichè ricerca:

- Esattezza, ovvero una descrizione precisa e non ambigua di cosa ci si debba aspettare da ogni costrutto sintatticamente corretto, affinchè l'utente sappia a priori quello che succederà durante l'esecuzione;
- Flessibilità, ovvero non deve anticipare scelte che possono essere demandate all'implementazione.

Per dare semantica ad un linguaggio dobbiamo sempre dare significato agli elementi complessi in funzione del significato degli elementi più semplici che lo compongono e del modo con cui questi sono composti.

[1] Per significato si intendono entità autonome che esistono indipendentemente dai segni che usiamo per descriverle -> es: la mano resta la mano, a prescindere da che termine uso per indicarla