

Introduzione ai Linguaggi di Programmazione

Sommario

- Linguaggi, messaggi e comunicazione
- Traduzione di programmi
- Interpreti e compilatori
- Introduzione al processo di compilazione

Linguaggio (1)

- Insieme di sequenze di *simboli, le parole*, appartenenti ad un definito *lessico*, giustapposti secondo una opportuna **grammatica** (o **sintassi**)
- Per descriverlo è necessario un **meta-linguaggio**
 - un linguaggio per descrivere un altro linguaggio

Linguaggio (2)

- In informatica un linguaggio è più precisamente definito come l'insieme di **stringhe** ottenute applicando le regole della relativa grammatica
- Di conseguenza, una grammatica è formalismo per definire un linguaggio mediante l'imposizione di un metodo per la costruzione delle stringhe

Linguaggio (3)

- Le stringhe ottenute possono essere analizzate da un punto di vista
 - Sintattico: Si verifica la forma linguistica in cui è codificato (**sintassi**)
 - Semantico: Si individua il significato associato alla forma linguistica (**semantica**)
 - Pragmatico: Si studiano gli usi e le utilità dei costrutti linguistici (**pragmatica**)

Comunicazione Diretta

- Requisiti per i due interlocutori:

l'estensore del messaggio (al momento della **formulazione**) e il ricevitore (alla **ricezione**) devono assegnare al messaggio **uguale significato**, ciascuno nel proprio contesto

Programma

- Messaggio **di comunicazione fra l'uomo e la macchina**
 - Insieme di frasi costruite secondo regole molto rigide
 - Eliminazione di ambiguità nell'interpretazione dei comandi da parte della macchina
 - Necessità di linguaggi molto precisi
 - Le istruzioni obbediscono a rigorose regole grammaticali

Comunicazione Indiretta

- È il caso della comunicazione tra il programmatore e il computer:
 - Il ricevitore non conosce il linguaggio usato per la stesura del messaggio
 - Estensore e ricevitore hanno un diverso grado di conoscenza del linguaggio
 - Tra i due mancano adeguate convenzioni per un'interpretazione unica del messaggio
- Occorre un **traduttore**

Linguaggio Naturale

- Usato per la comunicazione verbale fra esseri umani
 - Fonti di ambiguità:
 - Evoluzione
 - Neologismi, Arcaismi
 - Polisemia
 - Parole con significati differenti a seconda del contesto
 - Intrinseca
 - ...una vecchia porta la sbarra...
 - Inadatto alla comunicazione con la macchina
 - L'umano deve adattarsi a usare un linguaggio consono alla macchina

Linguaggi di Programmazione

- A basso livello
 - Più vicini alla struttura reale della macchina ed al suo linguaggio
- Ad alto livello
 - Più vicini al linguaggio dei problemi
 - Più facili da comprendere per l'uomo
 - Portabili
 - Utilizzabili, senza modifiche, su diversi tipi di macchine

Linguaggi di Programmazione ad alto livello

- Procedurali
 - Descrivono i passi necessari per ottenere i risultati desiderati
 - “come”
 - Basati sui concetti di
 - Variabile
 - Assegnamento
 - Es: Pascal, C, Ada, ...
- Non procedurali
 - Esprimono le proprietà dei risultati che si vogliono ottenere
 - “cosa”
 - Esempio Radice quadrata di y
 - Quel valore x tale che $x^2 = y$
 - $\text{sqrt}(x,y) :- y \text{ is } x^2$.
 - Es: Lisp, Prolog, ...

Linguaggi di Programmazione - Sintassi

- L'insieme delle regole che indicano quali sono le istruzioni formali permesse
 - Poche, semplici, rigide
- Il programma va accuratamente controllato dal punto di vista **formale** per garantire la correttezza sintattica
 - Codifica ambigua o non interpretabile
 - Controllo delegato al traduttore

Linguaggi di Programmazione - Semantica (1)

- Riguarda il contenuto informativo ed il significato di una frase
- Esistono metodi per trattare formalmente la semantica dei linguaggi di programmazione, al fine di capire il comportamento dei programmi

Linguaggi di Programmazione - Semantica (2)

- Semantica **Operazionale**: specifica come i (costrutti del) linguaggio sono eseguiti su una macchina astratta
- Semantica **Denotazionale**: definisce come denotare il significato dei (costrutti del) linguaggio
- Semantica **Assiomatica**: determina il significato dei (costrutti del) linguaggio usando regole di correttezza all'interno di una data logica

Sintassi e Semantica: Esempi

- La frase “Io ho andato”
 - È errata sintatticamente
- La frase “La penna sta mangiando”
 - È corretta sintatticamente (Forma), ma errata semanticamente (Significato)
- Le frasi “Tutti i bimbi sono buoni” e “Tutti i buoni sono bimbi”
 - Sono sintatticamente e semanticamente corrette
 - Hanno una semantica diversa nonostante siano costituite dagli stessi elementi

Traduttore

- Programma che traduce in linguaggio macchina programmi in un linguaggio di livello superiore
 - Analizza i messaggi (comandi) e verifica che siano scritti (codificati) in un linguaggio noto
 - Correttezza sintattica
 - Attribuisce alle sequenze di simboli l'opportuno significato in modo da eseguire le giuste azioni
 - Interpretazione unica di ogni istruzione
 - Fa parte del software di sistema
 - Livello intermedio della gerarchia software-hardware

Traduttori (1)

- Nei programmi ad alto livello operano su due tipi di entità:
 - Istruzioni
 - Molto più potenti che nel linguaggio macchina
 - Strutture di dati (sequenze, insiemi, alberi, ecc.)
 - Non direttamente disponibili al livello di linguaggio macchina
 - Devono essere rappresentate in termini di bit, indirizzi e legami tra locazioni

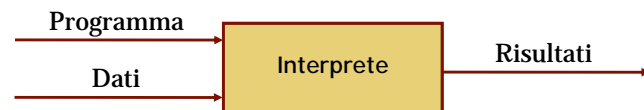
Traduttori (2)



- Interpreti
- Compilatori
 - Specifici per ogni linguaggio
 - Forniti entrambi dai sistemi di sviluppo del software per i linguaggi supportati

Interpretazione

- Dopo l'analisi sintattica, la traduzione procede passo passo con l'esecuzione
 - Traduzione ed esecuzione istruzione per istruzione
 - Ogni istruzione tradotta tante volte quante viene eseguita



Compilazione

- Il programma originale (Sorgente) è analizzato sintatticamente e tradotto in codice oggetto, quindi eseguito
 - Traduzione completamente effettuata prima che cominci l'esecuzione
 - Ogni istruzione è tradotta una sola volta



Interpreti vs. Compilatori

- | | |
|---|---|
| <ul style="list-style-type: none">• Programma sorgente• residente in memoria<ul style="list-style-type: none">– + Semplici– – Efficienti<ul style="list-style-type: none">• Tempo e Spazio– + Interattivi– + Errori comprensibili<ul style="list-style-type: none">• Riferiti al sorgente | <ul style="list-style-type: none">• Programma sorgente• non residente in memoria<ul style="list-style-type: none">– + Ottimizzabili– + Efficienti<ul style="list-style-type: none">• Tempo e Spazio– – Interattivi– + Errori scoperti prima<ul style="list-style-type: none">• Riferiti al codice oggetto |
|---|---|

Processo di Compilazione (1)

- **Analisi Lessicale (Scanning)**
 - Lettura in sequenza di tutti i caratteri che costituiscono il programma (il programma è visto come un'unica stringa di caratteri)
 - Divisione della stringa in token (segni di interpunzione, nomi di dati, operatori, parole riservate, ...)
 - Nessun controllo

Processo di Compilazione (2)

- **Analisi Sintattica (Parsing)**
 - Definizione della struttura sintattica del programma usando le regole del linguaggio
 - Un gruppo di token raggruppati secondo la strutturazione ottenuta deve corrispondere a una frase sintatticamente corretta secondo il linguaggio adottato
 - Se la frase non è sintatticamente corretta viene segnalato un messaggio d'errore

Processo di Compilazione (3)

- **Generazione del Codice**
 - Creazione di istruzioni in linguaggio macchina per ogni elemento sintattico del programma
 - L'insieme finale di queste istruzioni è il programma oggetto

Processo di Compilazione (4)

- Le fasi sono tra loro interrelate
 - I moduli di programma responsabili dell'analisi sintattica possono utilizzare
 - I moduli dell'analisi lessicale per ottenere un token
 - I moduli di generazione del codice per produrre il codice oggetto dell'istruzione analizzata
- Per capire come vengono effettuate le analisi è necessaria la teoria dei linguaggi