

Algoritmi e Strutture Dati

Stefania Monica

stefania.monica@unipr.it

Università degli Studi di Parma

A.A. 2019/2020

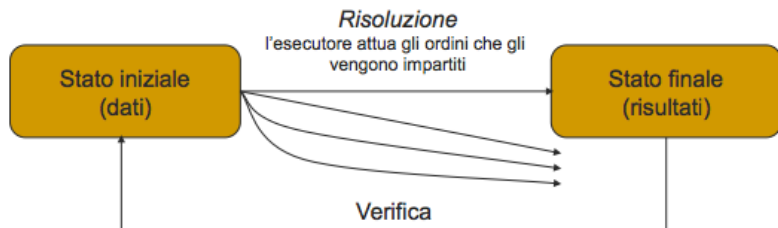
Definizione: Problema

Vediamo come viene definita la parola *problema* nei vocabolari.

- Larousse “Un problema è una ricerca che bisogna eseguire con procedimenti scientifici”.
- Devoto “Un problema è un quesito che richiede una soluzione. In matematica o in discipline affini è un quesito che richiede la determinazione o la costruzione di uno o più enti che soddisfino a date condizioni fissate in precedenza”.

Definizione: Soluzione

- Esprimere la soluzione di un problema vuol dire:
 - Partire da uno **stato iniziale** (insieme di dati iniziali);
 - Arrivare a uno **stato finale** (insieme dei dati finali o **risultati**);
 - Attraverso stati successivi intermedi (risultati parziali).
- La *soluzione* (o *procedimento risolutivo*) è l'insieme delle azioni che, applicate ai dati iniziali, consentono di pervenire ai risultati.
- I risultati saranno poi sottoposti alla verifica attraverso un opportuno **criterio di verifica**.



Domanda

Come possiamo risolvere un problema?

Definizione: Elaboratore

- Usiamo la parola *elaboratore* per indicare uno strumento che ci consente di esprimere un procedimento risolutivo.
- È un supporto (teorico e pratico) per esprimere la soluzione di problemi.
- Ogni elaboratore dipende dal problema di cui esprime la soluzione.

Domanda

Come possiamo risolvere un problema con un elaboratore?

- La soluzione si esprime attraverso azioni.
- Ci serve qualcosa in grado di compiere azioni.

Definizione: Esecutore

- Per risolvere un problema con un elaboratore, serve un *esecutore* che compia delle azioni che portino al risultato finale del problema.
- Ogni esecutore è definito dalle **azioni elementari** che può compiere.
- Un esempio di esecutore è il **calcolatore** (o **computer**).

Definizione: Calcolatore

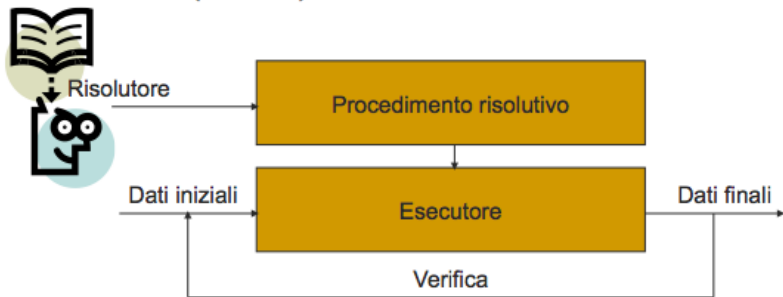
- Il *calcolatore* è un particolare tipo di esecutore.
- È uno strumento per la trasmissione, la trasformazione e la conservazione delle informazioni.
- Il calcolatore è indipendente dal problema, invece l'elaboratore esprime la soluzione a un problema dato.

Risolutore, Linguaggio e Azioni

- Il **risolutore** comunica le istruzioni da compiere all'esecutore usando un opportuno **linguaggio** compreso dall'esecutore.
- Le **istruzioni** fornite all'esecutore generano le **azioni**.

AZIONI = ISTRUZIONI + ESECUTORE

- I linguaggi sono il modo più semplice e completo per descrivere gli esecutori.



Definizione: Processo

- Un *processo* è un'**azione** composta da una sequenza di azioni elementari svolte da un esecutore.
- Le azioni elementari possono essere chiamate anche **passi**.

Definizione: Programma (o Procedura)

- Un *programma* è la descrizione di un processo, effettuata usando un linguaggio comprensibile dall'esecutore senza ambiguità.
- Quindi, un processo è un programma posto in esecuzione.

PROCESSO = PROGRAMMA + ESECUTORE

- La descrizione di un programma deve contenere:
 - L'elenco degli oggetti da manipolare, con il loro nome;
 - L'insieme delle azioni che devono essere compiute;
 - L'ordine in cui le azioni devono essere compiute, incluse le condizioni che devono essere verificate affinché a un'azione ne segua una o un'altra.

Algoritmo: Una Definizione Informale

Un *algoritmo* è un elenco **finito** di istruzioni **univocamente** interpretabili (da un esecutore), ciascuna delle quali deve essere definita in modo preciso, e la cui esecuzione **termina** per tutti i possibili dati iniziali validi, fornendo i rispettivi risultati.

Osservazioni

- La terminazione deve essere dimostrata (non possiamo provarlo davvero su tutti i processi).
- Possono esistere algoritmi diversi per risolvere lo stesso problema.
- Lo stesso algoritmo può risolvere un'intera classe di problemi.

Definizione: Algoritmo

Un *algoritmo* è un programma che, in tutti i processi che si possono creare, ha le seguenti caratteristiche:

- 1 Finitezza della descrizione;
- 2 Non limitatezza dei dati in ingresso;
- 3 Non limitatezza dei dati in uscita;
- 4 Non limitatezza del numero di passi eseguibili;
- 5 Definitezza;
- 6 Esistenza di un limite finito alla complessità delle istruzioni eseguibili;
- 7 Disponibilità per l'esecutore di memoria illimitata;
- 8 L'esecutore opera in modo discreto;
- 9 Terminazione per ogni dato in ingresso valido.

Qualche Osservazione...

- A volte servono programmi che non terminano.
 - Esempio: Programma che resta in ascolto dei messaggi sullo smartphone.
- Tecnicamente non si tratta di algoritmi, ma di *demoni* (termine tradizionale) o *agenti* (termine più moderno).
- Esempi di studio:
 - Sistemi operativi;
 - Web Server.

In Questo Corso...

Ci occupiamo di algoritmi e dovremo essere capaci di dimostrare sempre che i nostri programmi hanno le caratteristiche richieste per essere degli algoritmi.

Esecutori

In generale un esecutore ha ha disposizione:

- Un organo di ingresso;
- Un organo di uscita;
- Una memoria virtualmente illimitata;
- Una procedura risolutiva di un problema.

Quando un esecutore viene descritto in questo modo, si parla di *macchina astratta*.

Domanda

Parlando di esecutore, abbiamo parlato di linguaggio. Cos'è un linguaggio?

Definizione: Alfabeto

Un *alfabeto* è un insieme **non vuoto** e **finito** di simboli.

Definizione: Stringhe

Dato un alfabeto \mathcal{A} , l'insieme delle **sequenze finite** generabili con i simboli di \mathcal{A} si denota con \mathcal{A}^* .

Le sequenze vengo dette *stringhe*.

Definizione: Linguaggio

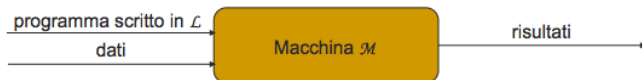
Un *linguaggio* \mathcal{L} su un alfabeto \mathcal{A} è un sottoinsieme di \mathcal{A}^* .

Nota: Linguaggio di Programmazione

Si parla di *linguaggio di programmazione* in riferimento a un linguaggio che ha lo scopo di dare istruzioni a una macchina.

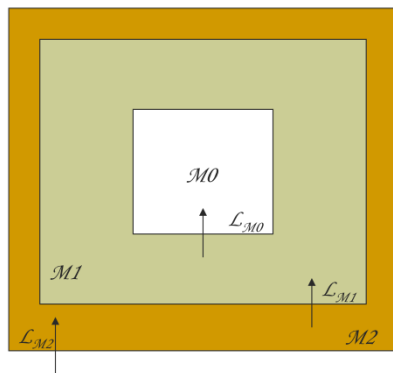
Linguaggi di Programmazione

- Un linguaggio \mathcal{L} è detto *linguaggio di programmazione* per una macchina \mathcal{M} se \mathcal{M} è un esecutore in grado di:
 - Decidere se una frase è ben formulata;
 - Eseguire i programmi scritti nel linguaggio \mathcal{L} .



Gerarchia di Macchine

- L'esecuzione di un programma scritto in linguaggio $\mathcal{L}_{\mathcal{M}_n}$ è affidata ad una catena arbitrariamente lunga di **macchine astratte** che si reggono su una **macchina concreta** \mathcal{M}_0 .



Ogni macchina è descritta completamente:

- Dal proprio linguaggio;
- Dalle proprie azioni elementari.

Interprete vs. Compilatore

Interprete

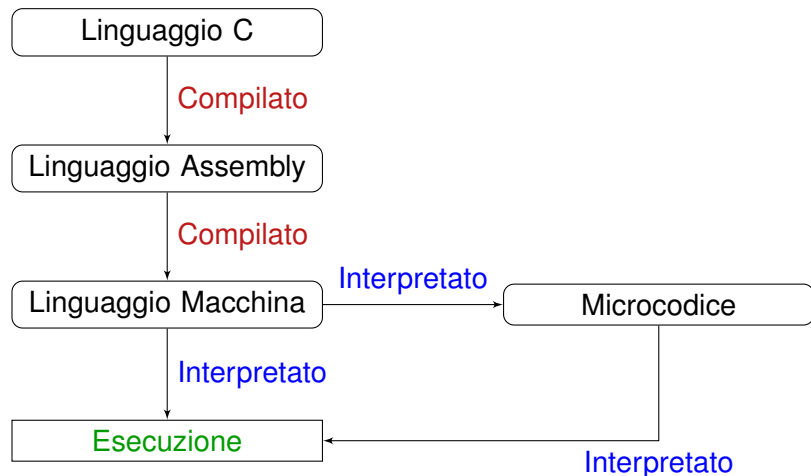
Esecuzione diretta degli ordini da parte della macchina.

- Ogni istruzione per \mathcal{M}_i viene immediatamente trasformata in una sequenza di istruzioni per \mathcal{M}_{i-1} .

Compilatore o Traduttore

- Realizza la traduzione di un programma nel linguaggio \mathcal{L}_i in un programma nel linguaggio \mathcal{L}_{i-1} , in modo semanticamente equivalente.
- Due fasi:
 - (*Compile-Time*) Traduzione di un programma P scritto nel linguaggio \mathcal{L}_i in un programma P' scritto nel linguaggio \mathcal{L}_{i-1} ;
 - (*Run-Time*) Esecuzione di P' da parte di \mathcal{M}_{i-1} .

Linguaggio C



Linguaggio Java

