UNIVERSITÀ DI GENOVA
FACOLTÀ DI SCIENZE M.F.N.

Dispense del corso di Laboratorio di Calcolo A
a.a. 2001-2002

Patrizia Boccacci, Paolo Morettini
Claudia Gemme, Maurizio Lo Vetere, Fabrizio Parodi

21st September 2001

CERN Program Library Office
CERN-CN Division
CH-1211 Geneva 23
Switzerland
Tel. +41 22 767 4951
Fax. +41 22 767 8630
Email: cernlib@cern.ch

Table of Contents

Capitolo 1: Introduzione

1.1 Che cos' è un programma

Il termine programma puó essere utilizzato per riferirsi a sequenze di azioni che si prefiggono il raggiungimento di certi obiettivi precisi: in un corso di studi il termine programma indica una sequenza temporale di argomenti da svolgere e di obiettivi didattici da raggiungere o in politica con termine programma di un partito si indicano una serie di azioni e di impegni da mantenere nell'immediato futuro.

Nonostante i numerosissimi casi in cui si utilizza il termine programma, uno degli usi piú frequenti è diventato quello riferito al mondo dei calcolatori.

DEF. Per **programma** si intende *una sequenza di istruzioni scritte in un linguaggio comprensibile al calcolatore* che le esegue per ottenere i risultati richiesti.

L'esecutore del programma è quindi il calcolatore. Nel 1642 il francese Blaise Pascal inventa una macchina da calcolo (chiamata pascalina) per aiutare il padre nel suo lavoro di funzionario delle imposte; il principio di funzionamento si basava sul movimento di ruote dentate.

All'inizio del Novecento le macchine calcolatrici divennero da meccaniche ad elettriche. Nel 1952 l'ungherese John von Neumann realizza il primo vero prototipo dei moderni elaboratori elettronici che poteva essere programmato per risolvere problemi diversi.

L'elaboratore entra nel mondo commerciale e da allora con una crescita esponenziale aumentano le prestazioni e si riducono le dimensioni e i costi di produzione.

Nel 1965 Gordon Moore (uno dei fondatori della Intel) formuló una legge che sosteneva che ogni 18 mesi sarebbe raddoppiato il numero dei transistor contenuti nei circuiti integrati. Questa legge, che è rimasta valida anche nel nuovo millennio con una piccola correzizone: da 18 mesi si e' passatiad un anno. (informazioni ulteriori http://www.intel.com/research/silicon/mooreslaw.htm)

Potete notare che il grafico inizia nel 1970, due anni dopo lo sbarco (presunto) dell'uomo sulla Luna.

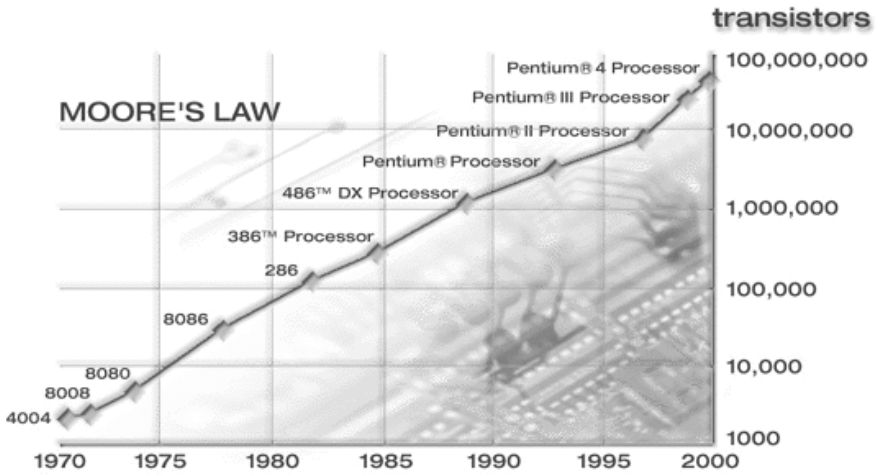
1.2 Che cos' è un algoritmo

Abbiamo appena accennato al concetto di programma nel campo degli elaboratori elettronici. Abbiamo detto che l'esecutore del programma è l'elaboratore elettronico e che il programma è una sequenza di istruzioni scritte in un linguaggio comprensibile al calcolatore. La *sequenza di istruzioni*, prima di essere tradotta in programma, rappresenta la *strategia* per raggiungere il risultato richiesto.

DEF. Si definisce **algoritmo** (dal nome del matematico arabo Al-Khuwarizmi) una serie finita di operazioni univocamente interpretabili, eseguite in sequenza, che trasforma i dati iniziali nel risultato richiesto.

Se il programma è la traduzione dell'algoritmo per poter essere eseguito senza errori dall'elaboratore questo deve essere univocamente interpretabile e deve essere rappresentato in una forma che ne consenta una facile conversione in programma.

L'insieme dei valori permessi dai dati in ingresso si definisce **dominio** dell'algoritmo (analogamente alle funzioni matematiche) mentre l'insieme dei valori che possono assumere i dati in uscita rappresenta il **codominio**.



In generale, ai fini di una migliore utilizzazione delle risorse disponibili (in termini di spazio e tempo) e per poter trattare problemi, anche di grandi dimensioni, con costi contenuti, è comunque fondamentale una scelta attenta nella formulazione dell'algoritmo risolutivo di un dato problema.

Una volta formulato l'algoritmo occorre verificare che questo fornisca:

- una soluzione per qualunque valore dei dati iniziali purchè appartenenti al dominio
- e che questa venga raggiunta in un numero finito di passi ovvero in un tempo finito.

E' chiaro che per descrivere un algoritmo funzionante devono essere considerati tutti i casi particolari anche con l'analisi dei risultati intermedi.

Una forma molto usata per la descrizione degli algoritmi è quella del *flow-chart*.

Esempio di algoritmo

Supponiamo di voler risolvere una equazione di secondo grado espressa in forma normale:

$$ax^2 + bx + c = 0 \tag{1.1}$$

Descriviamo questo semplice algoritmo prima a parole poi con un flow-chart.

1. Acquisisco a, b, c (supponiamo per semplicitá che a sia diversa da 0)
2. calcolo il $Delta = b^2 - 4ac$
3. Verifico se $Delta < 0$?
4. Se la risposta alla domanda precedente è *vero*, stampo che non ci sono soluzioni reali e vado al passo 10; se la risposta è *falso* passo al punto successivo.
5. Verifico se $Delta = 0$?
6. se la risposta alla domanda precedente è *vero*, pongo $x_1 = x_2 = -b/2a$, vado al passo 9; se la risposta è *falso* passo al punto successivo.
7. $x_1 = \frac{-b-\sqrt{\Delta}}{2a}$

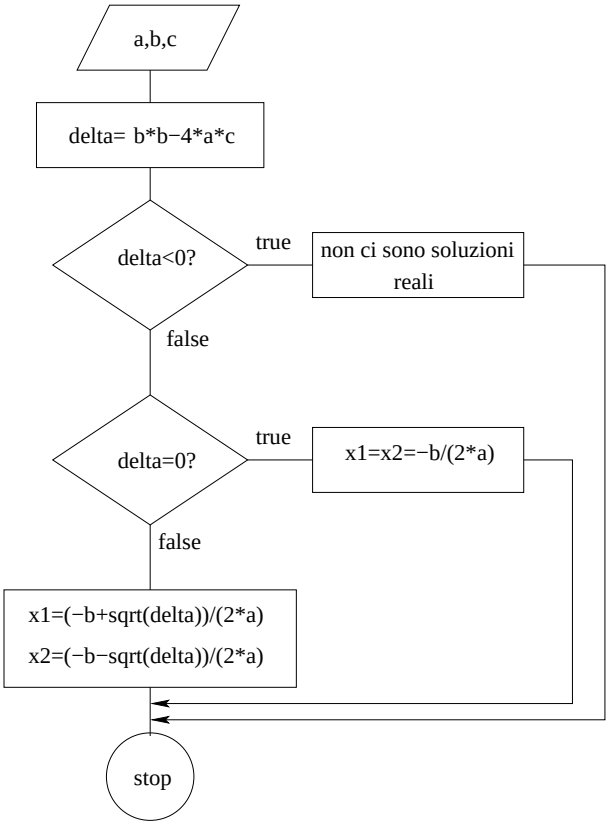


Figura 1.1: Grafico di flusso che descrive l'algoritmo per la risoluzione delle equazioni di secondo grado nel campo reale.

8. $x_2 = \frac{-b+\sqrt{\Delta}}{2a}$
9. stampo il risultato
10. termine della procedura