Appunti di informatica

Lezione 7

anno accademico 2016-2017

Mario Verdicchio

L’algoritmo di Euclide per l’MCD

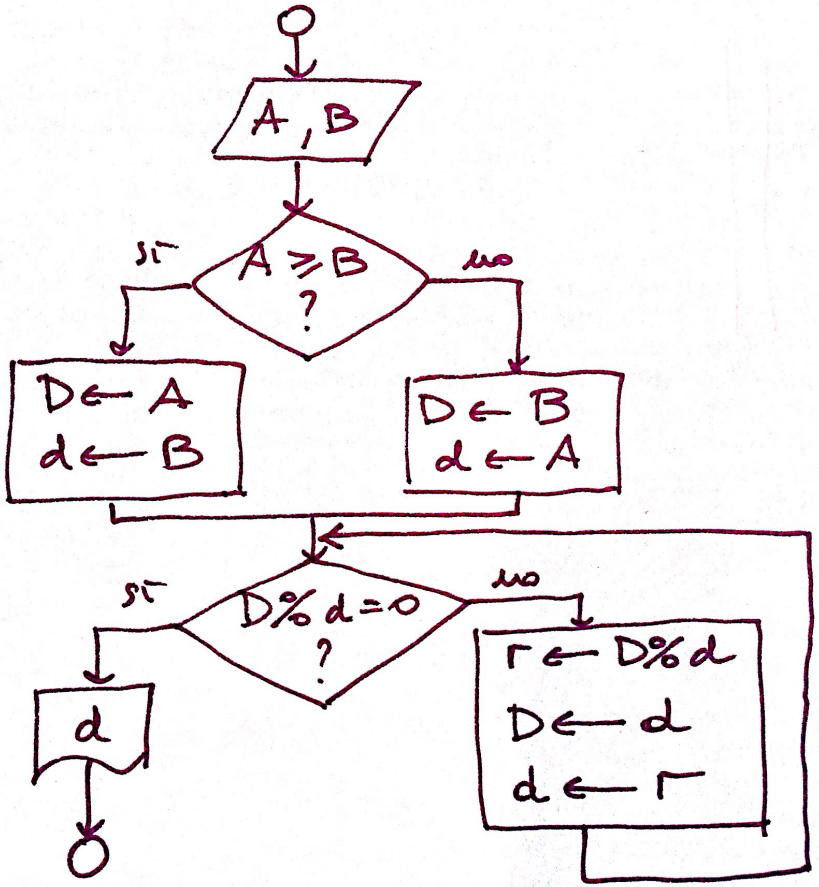
- Dati due numeri A e B, per trovare il loro MCD procedere nel seguente modo:
  1. dividere il maggiore per il minore
  2. se il resto è 0, il divisore è l’MCD
  3. altrimenti fare un’altra divisione: il vecchio divisore diventa il nuovo dividendo e il vecchio resto diventa il nuovo divisore, ripetere dal punto 2

Esempio con 150 e 70

dividendo	divisore	quoziente	resto
150	70	2	10
70	10	7	0

Dati 150 e 70 in input, alla prima divisione non otteniamo resto pari a 0, quindi ne eseguiamo un'altra con il divisore che diventa dividendo e resto che diventa divisore.
Alla seconda divisione il resto è 0, quindi il divisore è il MCD dei due numeri iniziali.
Infatti il MCD di 150 e 70 è 10.

Diagramma di flusso dell'algoritmo



Considerazioni

- L'algoritmo di Euclide e l'algoritmo per trovare l'MCD di due numeri visto nella lezione precedente sono la dimostrazione del fatto che, se esiste un algoritmo per risolvere un problema, è possibile che ve ne siano altri
- In realtà, ne esistono infiniti altri: basti pensare a modificare quello di Euclide aggiungendo istruzioni come r = r + n; r = r – n in un qualunque punto del diagramma (n può essere qualunque numero)
- Ovviamente alcune alternative presentano differenze significative (come quelle tra i due algoritmi proposti per l'MCD), mentre altre no (l'aggiunta di istruzioni inutili non modifica la soluzione in maniera sostanziale)

Un dubbio

- Abbiamo visto che per 150 e 70 l'algoritmo di Euclide funziona
- Chi ci garantisce che l'algoritmo funzioni per qualunque coppia di numeri in input?
- Definizione: un algoritmo si dice **corretto** quando risolve il problema per il quale è stato concepito
- Riformuliamo la domanda: chi ci garantisce che l'algoritmo di Euclide sia corretto?

Dimostrazione

- Innanzitutto una definizione
- **Dimostrazione**: sequenza finita di affermazioni tale che ogni affermazione è un'*ipotesi* presa per vera oppure deriva dalle affermazioni precedenti per mezzo di regole di inferenza (ragionamenti logicamente ineccepibili); l'ultima affermazione della sequenza si chiama *tesi*

Regole di inferenza

- Una regola di inferenza è un meccanismo con cui ottenere da una o più affermazioni una nuova affermazione. Ad esempio:

$$\frac{A \quad B}{A \wedge B}$$

- La regola “introduzione di congiunzione” dice che, a partire da A e B, possiamo ottenere $A \wedge B$.

Regole di inferenza corrette

- Una regola di inferenza si dice corretta se, a partire da affermazioni vere, ci permette di ottenere affermazioni vere
- La regola “introduzione di congiunzione” è corretta. Lo si può verificare mediante opportune tavole di verità
- Una regola come “eliminazione di disgiunzione”, invece, non è corretta:

$$\frac{A \vee B}{A}$$

  perché esiste un caso in cui $A \vee B$ è vera ma A è falsa
- A noi interessano solo regole di inferenza corrette
- La equivalente logiche sono un caso speciale di regole di inferenza

Uso delle regole di inferenza

- Le regole di inferenza si applicano a espressioni logiche
- Per poterle usare in una dimostrazione, quindi, dovremmo tradurre tutto il discorso in espressioni logiche
- In questo caso manteniamo il discorso in italiano e ci curiamo del fatto che i passaggi da un'affermazione alla successiva siano rigorosi

Dimostrazione di correttezza

- Dimostriamo che l'algoritmo di Euclide è corretto
- Nel caso in cui il resto della divisione tra A e B sia 0, è ovvio che il divisore B sia l'MCD, quindi in questo caso la correttezza è subito dimostrata
- Nel caso in cui il resto non sia 0, allora si passa a una nuova divisione: quella tra B e R
- Questa nuova divisione aiuta a risolvere il problema di trovare l'MCD tra A e B perché i divisori di A e B e i divisori di B e R sono in realtà lo stesso insieme

Dimostrazione di correttezza

- Per dimostrare che due insiemi sono uguali dobbiamo:
  1. dimostrare che un qualsiasi elemento del primo insieme appartiene al secondo insieme (cioè il primo insieme è un sottoinsieme del secondo)
  2. dimostrare che un qualsiasi elemento del secondo insieme appartiene al primo insieme (cioè il secondo insieme è un sottoinsieme del primo)
  3. l'unica possibilità per due insiemi che sono uno un sottoinsieme dell'altro è di essere coincidenti

I divisori di A e B sono divisori di B e R

- A:B = Q con resto di R
- ossia A = BQ + R, o anche R = A – BQ
- sia k un divisore di A e di B, ovvero esistono un m e un n tale che mk = A e nk = B
- questo vuol dire che R = mk – nkQ
- raccogliendo k, abbiamo che R = k(m – nQ)
- k, quindi, è divisore di B per ipotesi e, per quanto mostrato, è divisore anche di R
- perciò k è divisore di B e di R

I divisori di B e R sono divisori di A e B

- Sappiamo già che $A = BQ + R$
- sia h un divisore di B e di R, ovvero esistono un s e un t tale che $sh = B$ e $th = R$
- questo vuol dire che $A = shQ + th$
- raccogliendo h, abbiamo che $A = h(sQ + t)$
- h, quindi, è divisore di B per ipotesi e, per quanto mostrato, è divisore anche di A
- perciò k è divisore di A e di B

Dimostrazione di correttezza

- Il fatto che i due insiemi coincidano vuol dire che la soluzione del problema MCD per A e B è la stessa del problema MCD per B e R
- Applicando lo stesso ragionamento, sappiamo che la soluzione non cambia nemmeno per le divisione successive
- Non appena troviamo che il resto di una divisione è 0, sappiamo che il divisore è l'MCD della coppia dividendo-divisore
- Per quanto detto prima, questo sarà l'MCD della coppia iniziale di numeri A e B

Avvertenze

- Si è riuscito a dimostrare la correttezza dell'algoritmo di Euclide grazie alle proprietà matematiche del problema affrontato
- Non è scontato che si riesca a dimostrare sempre così facilmente la correttezza di un algoritmo
- Inoltre, anche se si ha un algoritmo di correttezza dimostrata, trasformandolo in programma (cioè riscrivendolo in un linguaggio di programmazione), il programmatore umano potrebbe inserire numerosi errori

Esercizio

- Disegnare il diagramma di flusso di un algoritmo che funziona come segue: riceve in input un numero x, e manda in output, in ordine decrescente, tutti i suoi divisori.