



PEGASO
Università Telematica



Indice

| | |
|---|----|
| 1. RIEPILOGO DELLA PROGRAMMAZIONE STRUTTURATA | 3 |
| RIFERIMENTI BIBLIOGRAFICI | 11 |

1. Riepilogo della programmazione strutturata

Abbiamo imparato che la programmazione strutturata produce programmi più facili da capire rispetto ai programmi non strutturati e quindi più facili da verificare, correggere, modificare e persino dimostrare in senso matematico come corretti.

Fino ad ora ci siamo concentrati sulle istruzioni di controllo del C.

Ogni istruzione è stata presentata, rappresentata con diagrammi di flusso ed esaminata separatamente con esempi.

Per chiarezza è opportuno riepilogare quanto visto fino ad ora e introdurre un insieme semplice di regole per la costruzione e la verifica di proprietà dei programmi strutturati.

La Figura 1 seguente riepiloga le istruzioni di controllo esaminate fino ad ora.

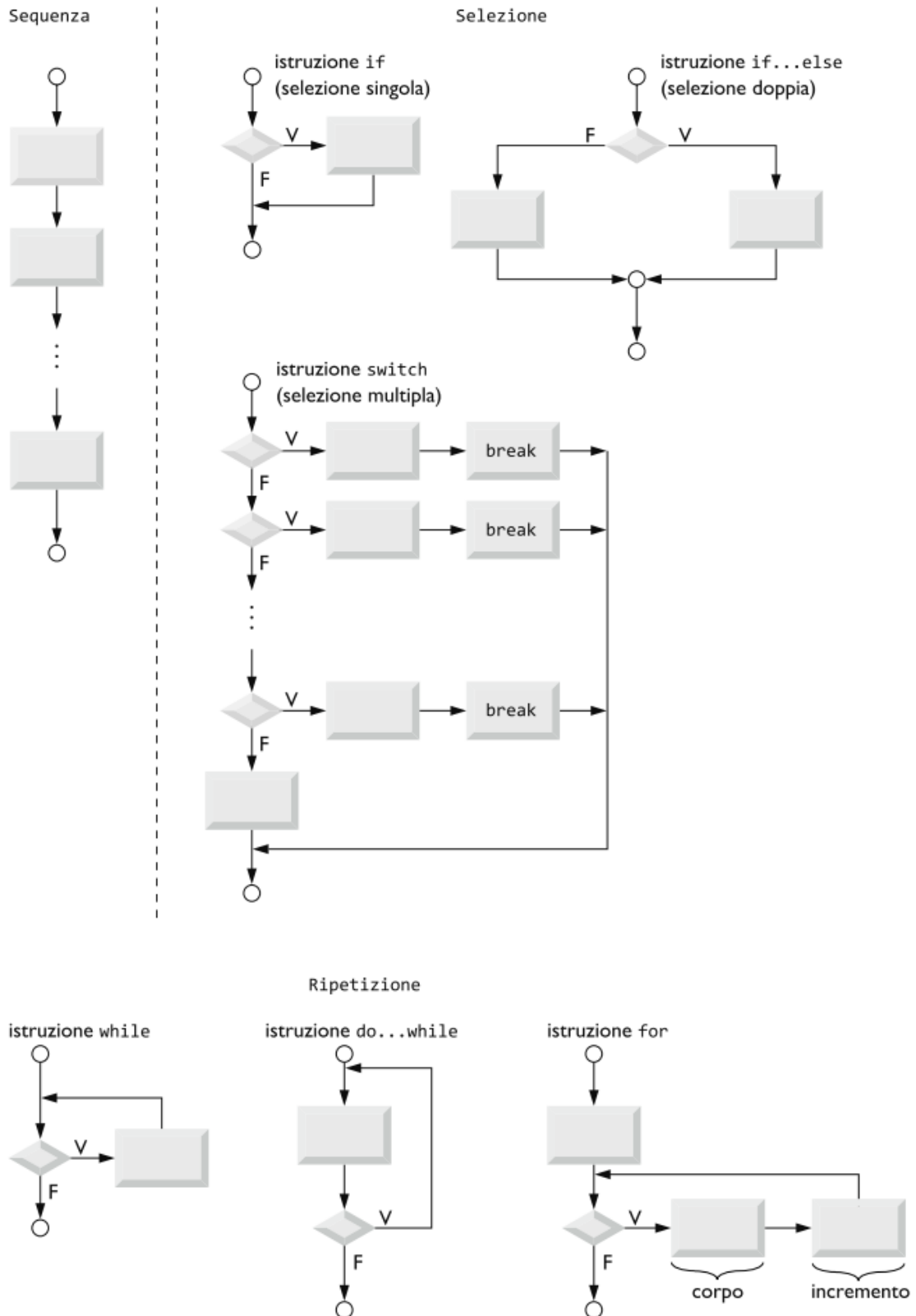


Figura 1. Istruzioni in C a un solo ingresso e a una sola uscita per sequenza, selezione e iterazione.

I cerchietti sono usati nella figura per indicare l'unico punto di entrata e l'unico punto di uscita di ciascuna istruzione.

Collegare arbitrariamente simboli individuali dei diagrammi di flusso può portare a programmi non strutturati.

Pertanto, la professione della programmazione ha scelto di combinare i simboli dei diagrammi di flusso in un insieme limitato di istruzioni di controllo e di costruire soltanto programmi strutturati opportunamente combinando correttamente le istruzioni di controllo in due semplici modi.

Per semplicità sono usate solo istruzioni di controllo con un solo ingresso e a una sola uscita: c'è solo un modo per entrare e solo un modo per uscire da ogni istruzione di controllo.

Collegare istruzioni di controllo in sequenza per formare programmi strutturati è semplice (il punto di uscita di un'istruzione di controllo è collegato al punto di entrata della successiva); le istruzioni di controllo sono, cioè, semplicemente poste una dopo l'altra in un programma.

Abbiamo chiamato ciò "accatastamento delle istruzioni di controllo".

Le istruzioni di controllo possono anche essere annidate.

La Figura 2 mostra le regole per formare programmi strutturati.

Le regole presuppongono che il simbolo rettangolo dei diagrammi di flusso possa essere usato per indicare qualsiasi azione, incluso l'input/output.

La Figura 3 mostra il diagramma di flusso più semplice.

Applicando le regole della Figura 2 si produce sempre un diagramma di flusso strutturato che si presenta in modo chiaro come un blocco costituente.

Applicando ripetutamente la Regola 2 al diagramma di flusso più semplice (Figura 3) si produce un diagramma di flusso strutturato contenente molti rettangoli in sequenza (Figura 4).

La Regola 2 genera una pila di istruzioni di controllo, e pertanto la chiamiamo regola di accatastamento.

La Regola 3 è chiamata regola di annidamento.

Applicando ripetutamente la Regola 3 al diagramma più semplice, si produce un diagramma di flusso con le istruzioni di controllo annidate in maniera evidente.

Ad esempio, nella Figura 5 il rettangolo nel diagramma di flusso più semplice è dapprima sostituito con un'istruzione di selezione doppia (`if...else`).

Dopodiché, si applica di nuovo la Regola 3 a tutti e due i rettangoli nell'istruzione di selezione doppia, sostituendo ognuno di essi con istruzioni di selezione doppia.

La scatola tratteggiata attorno a ognuna delle istruzioni di selezione doppia rappresenta il rettangolo che è stato sostituito nel diagramma di flusso originario.

La Regola 4 genera strutture più ampie, più complicate e annidate più in profondità.

I diagrammi di flusso che risultano dall'applicazione delle regole della Figura 2 costituiscono l'insieme di tutti i possibili diagrammi di flusso strutturati e, di conseguenza, l'insieme di tutti i possibili programmi strutturati.

È a causa dell'eliminazione dell'istruzione `goto` che questi blocchi costituenti non si sovrappongono mai l'uno sull'altro.

La bellezza dell'approccio strutturato è che noi usiamo soltanto un piccolo numero di pezzi semplici a un solo ingresso e a una sola uscita e possiamo assemblarli in due soli semplici modi.

La Figura 6 mostra i tipi di blocchi costituenti accatastati che risultano dall'applicazione della Regola 2 e i tipi di blocchi annidati che risultano dall'applicazione della Regola 3.

La figura mostra anche il tipo di blocchi costituenti sovrapposti che non possono comparire in diagrammi di flusso strutturati (a causa dell'eliminazione dell'istruzione `goto`).

Se si seguono le regole della Figura 2, non si può creare un diagramma di flusso non strutturato (come quello nella Figura 7).

Se non siete certi che un particolare diagramma di flusso sia strutturato, applicate inversamente le regole della Figura 2 per cercare di ridurre il diagramma di flusso al diagramma di flusso più semplice.

Se ci riuscite, il diagramma originario è strutturato, altrimenti non lo è.

Regole per costruire programmi strutturati:

1. Iniziare con il "diagramma di flusso più semplice"
2. Regola di "accatastamento": Qualsiasi rettangolo (azione) può essere sostituito con due rettangoli (azioni) in sequenza.
3. Regola di "annidamento": Qualsiasi rettangolo (azione) può essere sostituito con qualunque istruzione di controllo (sequenza, if, if...else, switch, while, do...while o for).
4. Le regole 2 e 3 possono essere applicate quante volte si vuole e in qualsiasi ordine.

Figura 2. Regole per costruire programmi strutturati.

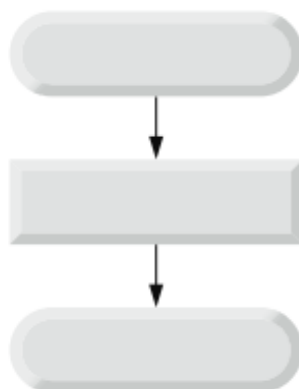


Figura 3. Diagramma di flusso più semplice.

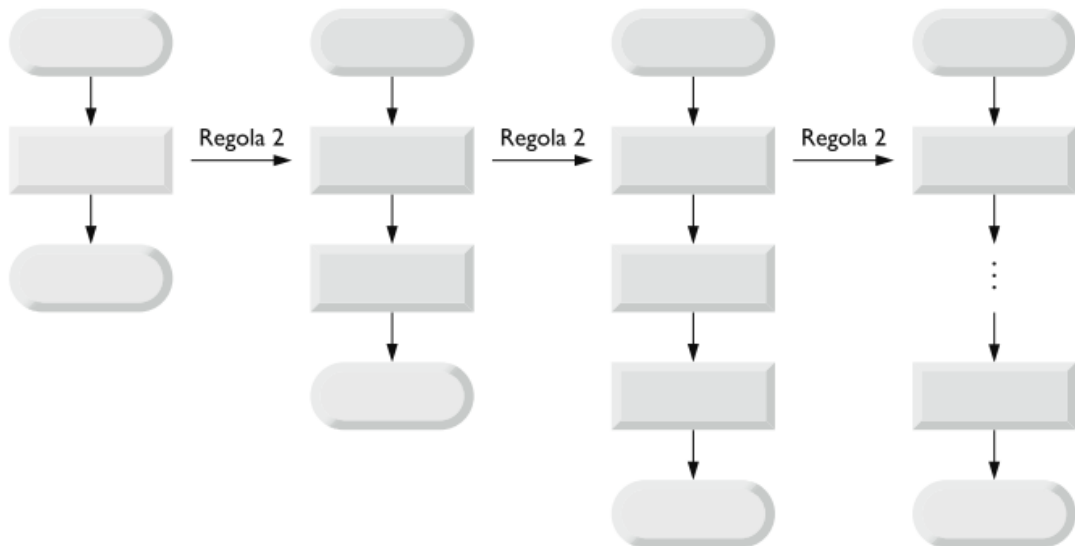


Figura 4. Applicazione ripetuta della Regola 2 della Figura 2 al diagramma di flusso più semplice.

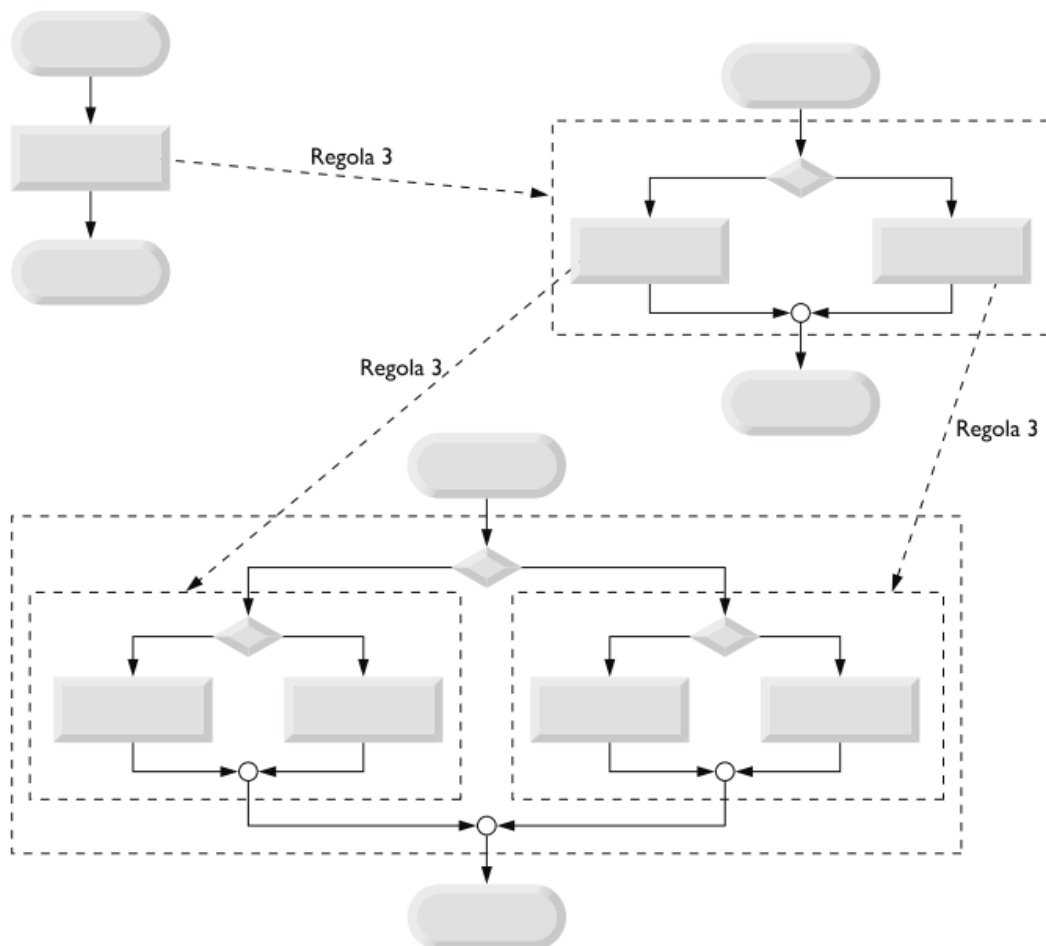


Figura 5. Applicazione della Regola 3 della Figura 2 al diagramma di flusso più semplice.

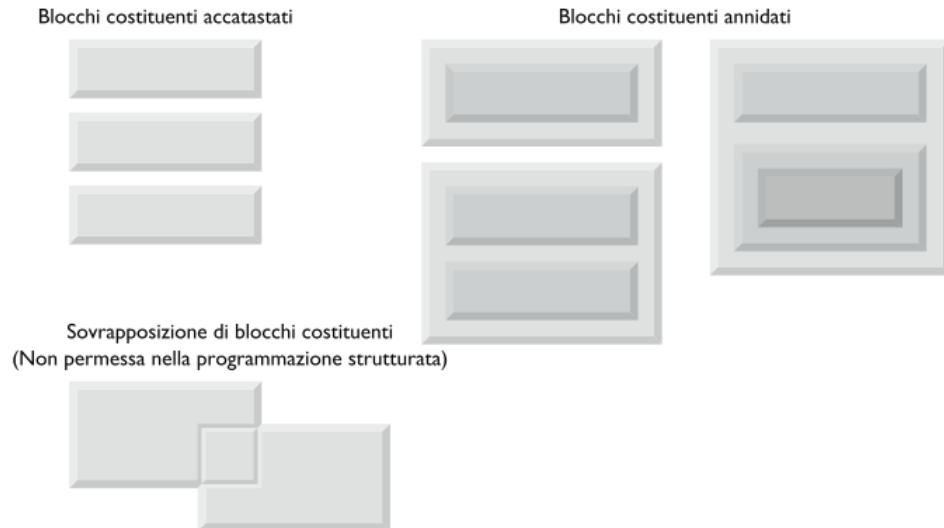


Figura 6. Blocchi costituenti accatastati, annidati e sovrapposti.

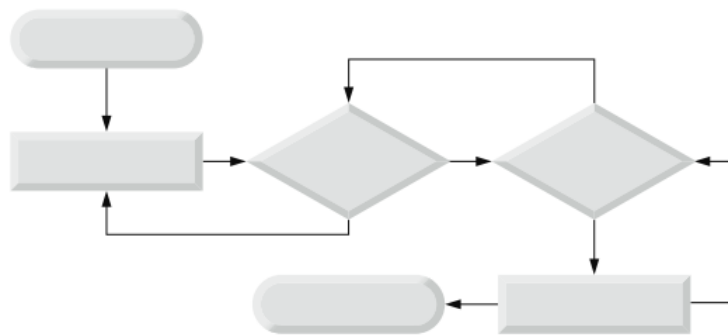


Figura 7. Un diagramma di flusso non strutturato.

La programmazione strutturata favorisce la semplicità.

E' stato dimostrato che sono necessarie solo tre forme di controllo:

- Sequenza
- Selezione
- Iterazione

La sequenza è lineare.

La selezione è implementata in uno dei tre modi:

- istruzione if (selezione singola)
- istruzione if...else (selezione doppia)

- istruzione switch (selezione multipla)

È facile dimostrare che la semplice istruzione if è sufficiente a realizzare qualunque forma di selezione, ossia tutto quello che si può fare con l'istruzione if...else.

Inoltre, l'istruzione switch può essere implementata con una o più istruzioni if.

L'iterazione è implementata in uno dei tre modi:

- istruzione while
- istruzione do...while
- istruzione for

È inoltre facile dimostrare che l'istruzione while è sufficiente per ottenere qualunque forma di iterazione.

Tutto ciò che può essere fatto con l'istruzione do...while e con l'istruzione for può essere fatto con l'istruzione while.

La combinazione di questi risultati dimostra che qualsiasi forma di controllo che possa mai risultare necessaria in C può essere espressa in termini di solo tre forme di controllo:

- sequenza
- istruzione if (selezione)
- istruzione while (iterazione)

E queste istruzioni di controllo possono essere combinate in due soli modi: accatastamento e annidamento.

La programmazione strutturata favorisce davvero la semplicità.

Riferimenti bibliografici

- Paul Deitel, Harvey Deitel, "Il linguaggio C – Fondamenti e tecniche di programmazione",
Libro edito da Pearson Italia. Include anche utili esercizi di autovalutazione.