

pletamente fu dimostrata formalmente dalla ricerca di due matematici italiani: Corrado Böhm e Giuseppe Jacopini, che provarono [7] che il flusso di qualunque programma si può sempre scrivere in termini di tre sole *strutture di controllo*: esecuzione sequenziale, selezione e iterazione.

1. Nella struttura sequenziale il programma si sviluppa eseguendo le istruzioni nell'ordine in cui sono scritte. Si tratta, in questo caso, del normale modo di procedere.
2. Con la struttura di selezione il programma raggiunge un punto in cui si valuta un'espressione logica il cui risultato (vero o falso) determina la successiva esecuzione di una parte delle istruzioni.
3. La struttura di iterazione permette di eseguire ciclicamente una serie di istruzioni identiche, fino a che non si verifichino certe condizioni.

Le strutture di controllo si rappresentano come diagrammi di flusso che hanno un unico punto d'ingresso e un unico punto d'uscita. Il teorema di Böhm-Jacopini garantisce che qualunque programma, non importa quanto complesso, si può riscrivere in termini di tre sole strutture di controllo e che dunque non c'è alcun bisogno di eseguire un salto da un'istruzione all'altra: in questo caso il programma si dice *strutturato*. Nonostante questo, buona parte dei linguaggi di programmazione esistenti (incluso il C) consente di scrivere programmi non completamente strutturati. Un programma strutturato consente di controllare meglio il flusso del codice rispetto a uno non strutturato e per questo è buona norma cercare di usare, per quanto è possibile, questo tipo di programmazione. Come spesso accade, l'adesione rigorosa a un sistema di regole rigide può provocare, talvolta, difficoltà di altra natura<sup>1</sup>. Per questa ragione derogheremo, in alcuni casi, dalle prescrizioni del teorema di Böhm-Jacopini, ma questo potrà avvenire solo in situazioni molto specifiche. In particolare esistono istruzioni come **break**, **continue** e **switch** che sono oggetto di disputa relativamente al fatto se siano o meno impiegabili in un programma strutturato. Noi non entriamo in questa disputa, ma ne evitiamo l'uso.

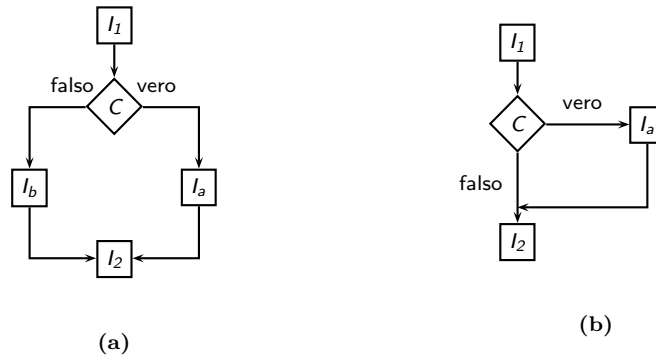
## 4.2 Prendere una decisione

La struttura di selezione individuata dal teorema di Böhm-Jacopini permette di scegliere la porzione di codice da eseguire secondo il valore di verità di una proposizione. Se chiamiamo  $I_a$  e  $I_b$  due gruppi di istruzioni mutuamente esclusive e indichiamo con  $C$  la proposizione da verificare, la struttura si può rappresentare simbolicamente con il diagramma di flusso nella Figura 4.2 (a).

Il programma comincia con le istruzioni  $I_1$ ; successivamente valuta il valore di verità della proposizione  $C$ : se tale valore risulta vero si esegue il gruppo di istruzioni  $I_a$ , altrimenti si passa all'esecuzione del blocco  $I_b$ . Al termine dell'esecuzione dell'uno o dell'altro insieme di istruzioni il programma prosegue eseguendo le istruzioni  $I_2$ . Il diagramma di

---

<sup>1</sup> In alcuni casi l'uso di istruzioni non strutturate può portare a una maggiore efficienza del programma. Questi casi sono estremamente rari perché i compilatori moderni riescono a ottimizzare le prestazioni del codice in modo molto efficiente. In altri casi il codice strutturato può apparire meno chiaro.



**Figura 4.2** La struttura di selezione piú generale (a) e quella piú semplice (b), che si ottiene da quella fondamentale sostituendo le istruzioni  $I_b$  con l'istruzione nulla.

flusso nella Figura 4.2 (a) si può tradurre in un linguaggio informale come: «Se  $C$  è vera esegui  $I_a$ , altrimenti esegui  $I_b$ ».

La struttura 4.2 (a) è quella fondamentale: da essa si generano facilmente le altre. In alcune circostanze, ad esempio, l'insieme delle istruzioni  $I_b$  è vuoto, e in questi casi si genera una struttura di selezione piú semplice, priva di alternative, come quella illustrata in 4.2 (b). In questo caso l'esecuzione del programma comincia dalle istruzioni  $I_1$ , quindi si valuta il valore della proposizione  $C$ . Se tale valore risulta vero, prima di passare all'esecuzione delle istruzioni  $I_2$ , vengono eseguiti i comandi in  $I_a$ , altrimenti si passa direttamente all'esecuzione di  $I_2$ .

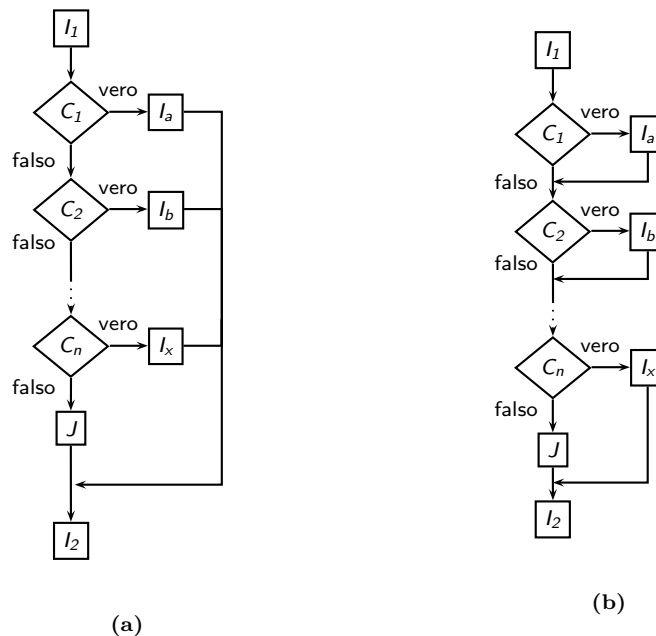
Componendo piú strutture di selezione si possono ottenere costrutti per fare scelte piú complesse, del tipo di quella illustrata nella Figura 4.3 (a). Si tratta semplicemente dalla concatenazione di strutture come quelle mostrate nella Figura 4.2 (a), in cui l'uscita di una struttura è connessa all'ingresso della successiva in modo tale da sostituire il blocco  $I_b$  con una struttura di selezione. Tra le istruzioni  $I_1$  e  $I_2$  si esegue un insieme di istruzioni  $I_k$  che dipende dal verificarsi di una tra le condizioni  $C_j$ . Nel caso in cui tutte le condizioni risultino false si eseguono comunque le istruzioni  $J$  (l'insieme di queste istruzioni può essere vuoto). Questa struttura corrisponde a quella generata dall'uso dell'istruzione `switch` del linguaggio C (Paragrafo 4.4). Un'altra possibilità è costituita dalla concatenazione di strutture non mutuamente esclusive, generata dalla successione delle strutture di selezione di Figura 4.2 (b), riportata nella Figura 4.3 (b).

La differenza tra le due strutture è solo apparentemente minima: l'esecuzione del programma può cambiare drasticamente. Nel caso 4.3 (a) il verificarsi di una condizione esclude la possibilità che se ne verifichino altre e, una volta eseguite le istruzioni corrispondenti, l'esecuzione del programma passa immediatamente alle istruzioni  $I_2$ . Nel caso 4.3 (b), invece, il verificarsi di una condizione non esclude la possibilità che se ne verifichino altre. Questo comporta che piú gruppi di istruzioni  $I_k$  possano essere eseguiti

ESTRATTO DAL LIBRO **PROGRAMMAZIONE SCIENTIFICA** DI

L. M. BARONE, E. MARINARI, G. ORGANTINI E F. RICCI-TERSENGHI

I diritti di riproduzione e memorizzazione elettronica degli estratti appartengono a Pearson Education Italia S.r.l. e sono riservati per tutti i paesi. La stampa di essi è concessa gratuitamente solo a titolo personale. Ogni altro uso è vietato.



**Figura 4.3** Concatenando diverse strutture di selezione si possono ottenere strutture a scelta multipla: nella struttura a sinistra le scelte sono tra loro mutuamente esclusive, mentre in quella a destra una scelta non esclude le altre.

e che, mentre nel primo caso si determina il valore delle sole condizioni da  $C_1$  a quella risultata vera, nell'altro tutte le condizioni vengono comunque valutate. Se dunque le condizioni sono tra loro mutuamente esclusive, occorre sempre scegliere un costrutto come quello mostrato nella Figura 4.3 (a) e, per migliorare le prestazioni del codice, è opportuno inserire prima le condizioni che hanno maggiore probabilità di verificarsi.

### 4.2.1 if/else

La struttura di selezione si può realizzare nel linguaggio C attraverso il costrutto `if`, come mostriamo nel Listato 4.1.

---

```

18 printf("Valore in gradi celsius= ");
19 scanf("%lf", &tc);
20 if (tc < -273.16) printf("Attenzione! T<0 K...\n");

```

---

**Listato 4.1** Un esempio di uso dell'istruzione `if`.

---

ESTRATTO DAL LIBRO **PROGRAMMAZIONE SCIENTIFICA** DI  
L. M. BARONE, E. MARINARI, G. ORGANTINI E F. RICCI-TERSENGHI  
I diritti di riproduzione e memorizzazione elettronica degli estratti appartengono a  
Pearson Education Italia S.r.l. e sono riservati per tutti i paesi. La stampa di essi è  
concessa gratuitamente solo a titolo personale. Ogni altro uso è vietato.