

Introduzione

A partire dalla metà degli Anni Quaranta, con la creazione e la costruzione di numerosi calcolatori fra loro eterogenei – sia per caratteristiche tecniche che per finalità d’uso, potenzialità di calcolo e contesti in cui avvenne la loro progettazione – iniziò a svilupparsi l’idea secondo cui fosse necessario concepire nuove modalità di interazione fra le macchine e gli operatori che ne facevano uso, fossero essi gli ingegneri costruttori specializzati o, successivamente e con l’espansione del mercato informatico, i tecnici indipendenti non formati nei laboratori delle ditte produttrici e di conseguenza dotati di una preparazione non vincolata a una singola tipologia di calcolatori ma potenzialmente di più alto livello e respiro. Ci si preoccupò, quindi, di ricercare un approccio più intuitivo e *human-like* alla programmazione delle macchine calcolatrici, la maggior parte delle quali fino a quel momento prevedeva che l’unica maniera per settare il valore di ogni singolo bit coinvolto in un calcolo fosse rappresentata da un intervento fisico dell’uomo, finalizzato a variare di volta in volta le connessioni fra i diversi circuiti o lo stato di particolari interruttori o componenti.

Durante il percorso che portò alla concezione di tecniche e modalità di programmazione più semplici, veloci e versatili – dalle quali, in seguito, sarebbero sorti i primi veri linguaggi di programmazione a larga diffusione – assunsero particolare interesse alcune specifiche innovazioni, tecnologiche da una parte e teoriche e metodologiche dall’altra, fra cui:

- l’introduzione di periferiche di input, quali per esempio interpreti di schede perforate o pannelli di controllo, utilizzate come interfacce fra l’operatore e i circuiti veri e propri del calcolatore;

- la rappresentazione delle istruzioni tramite codifiche mnemoniche (per esempio ADD) e non decimali o binarie;
- la rappresentazione decimale degli operandi anche nel caso in cui essi venissero trattati dalla macchina con codifica binaria;
- la possibilità di scrivere gli algoritmi in pseudo-codici di facile interpretazione.

Questa tesi si colloca nell'ampio panorama della ricostruzione storica dell'evoluzione dell'informatica e si prefigge, in particolare, di fare luce su uno dei momenti-chiave dello sviluppo della programmazione che ancora non è stato trattato a fondo e in maniera del tutto organica e completa. Scopo di questo lavoro è fornire una panoramica dettagliata dei linguaggi più rilevanti fra quelli specifici degli elaboratori progettati e costruiti fra la metà degli Anni Quaranta e la metà degli Anni Cinquanta, mettendo in evidenza il radicale cambiamento rispetto alla precedente programmazione binaria e la progressiva evoluzione in termini di versatilità e potenzialità che è avvenuta con il passaggio fra ciascuno di questi linguaggi e il successivo.

La scelta dei calcolatori su cui incentrare la discussione è caduta su ENIAC, EDSAC, EDVAC, IBM 650 e Bendix G-15, in quanto essi coprono un ampio ventaglio delle macchine significative disponibili nel decennio preso in considerazione, sia per caratteristiche tecniche dei componenti e specificità dei propri sistemi di calcolo che per peculiarità relative ai luoghi e alle circostanze che li videro nascere, nonché per i differenti ambiti di applicazione a cui vennero destinati. Se le prime furono macchine disponibili in un unico esemplare ed ebbero finalità riguardanti le sole sfere militare, universitaria e di ricerca scientifica, gli ultimi due modelli oggetto di questa trattazione vennero costruiti in serie per l'allora nascente mercato civile dell'informatica e godettero di diffusione su media o larga scala a livello internazionale.

Per ognuna delle macchine trattate si è cercato in questa tesi di descrivere inoltre il contesto che portò alla sua costruzione, l'architettura che la caratterizzava, le eventuali migliorie a essa apportate nel tempo e i pregi e i difetti riscontrabili confrontandone le potenzialità con quelle dei calcolatori dello stesso periodo.

La tesi è divisa in sei capitoli, i cui titoli e contenuti sono qui elencati e riassunti:

1. **Modelli teorici.** Viene esposta la nozione fondamentale di *Macchina di Turing*, se ne forniscono le definizioni informale e formale e viene valutata la portata di tale concetto. Inoltre, viene presentato e descritto nel dettaglio il linguaggio *Plankalkül*, concepito da Konrad Zuse ma rimasto solo teorico, e se ne esaminano pregi e difetti alla luce della vera evoluzione della storia della programmazione;
2. **ENIAC.** Si passa poi alla trattazione dell'*Electronic Numerical Integrator and Computer*, ovvero il primo in ordine cronologico fra gli elaboratori di cui ci si occupa in questo lavoro. Di tale macchina vengono analizzate la prima versione, che non consentiva una programmazione agevole, e l'evoluzione successiva, che prevedeva invece l'utilizzo di particolari istruzioni che semplificavano il lavoro dell'operatore;
3. **EDSAC ed EDVAC.** Il terzo capitolo illustra il fattore comune che portò alla nascita dell'*Electronic Delay-Storage Automatic Calculator* e dell'*Electronic Discrete Variable Automatic Computer* e si preoccupa poi di evidenziare le peculiarità di tali due macchine, con particolare riguardo all'analisi dei due diversi set di istruzioni di cui esse erano dotate e alle differenti modalità di programmazione che ne conseguivano;
4. **IBM 650 e SOAP II.** In questo capitolo vengono descritte le caratteristiche tecniche e strutturali dell'*IBM 650 Magnetic Drum Calculator* e viene presentato e analizzato nel dettaglio il linguaggio di assemblaggio a esso associato e denominato *Symbolic Optimal Assembly Program II*. Si forniscono, infine, due esempi di programmi creati con tale linguaggio;
5. **Bendix G-15.** Nel quinto capitolo viene illustrata l'architettura del Bendix G-15 e viene posto l'accento sulla particolare struttura dei suoi comandi e sul fatto che essi potessero essere modificati e ridefiniti a seconda delle esigenze dell'operatore;
6. **Tavole comparative.** Vengono fornite, infine, alcune tabelle per la rapida comparazione delle informazioni relative alla storia, all'architettura, alla memoria e alla metodologia di programmazione delle macchine prese in esame.