

Non solo un traduttore di formule: Il FORTRAN e i suoi 60 anni

C. Petrocelli

Sommario. Il 10 novembre del 1954, IBM divulgò un rapporto intitolato “Specifications for the IBM Mathematical FORMula TRANslating System, FORTRAN” che segnò la nascita di un linguaggio di programmazione ancora oggi utilizzato. A distanza di 60 anni si ripercorrono le tappe della sua diffusione.

Keywords: John Warner Backus, FORTRAN, IBM, History of programming languages

1. Gli anni '50: Programmazione sinonimo di pseudocodifica

Il periodo compreso tra la fine della II Guerra Mondiale e la metà degli anni Cinquanta, fu caratterizzato fondamentalmente dalla nascita dei primi calcolatori elettronici a programma memorizzato, secondo i modelli proposti da Turing e von Neumann, per i quali la memoria della macchina avrebbe dovuto essere utilizzata per le elaborazioni delle istruzioni dei programmi e contemporaneamente dei dati in esse coinvolti. Ma, in quegli anni, la *programmazione* era per lo più un'arte misteriosa, una materia arcaica che solo in pochi avevano svelato facendo uso di tecniche ed inventiva che certo non garantivano leggibilità ed adattabilità dei programmi stessi. Questi ultimi erano infatti funzionanti solo ed esclusivamente sulla macchina per cui erano stati codificati e venivano impostati secondo il criterio personale del programmatore il quale, spesso, doveva far uso della 'soluzione del momento' per la produzione di un codice efficiente.

L'inevitabile ridondanza che ne era conseguita, aveva cominciato a far riflettere sulla necessità di definire e formalizzare degli standard al fine di ottimizzare gli sforzi di ogni programmatore, consentendogli di concentrarsi solo ed

esclusivamente sul problema da risolvere. Bisognava pertanto, in primo luogo semplificarne il processo di scrittura utilizzando, ad esempio, un linguaggio di più alto livello rispetto al codice macchina e contemporaneamente rendere il codice 'usabile', cioè standardizzarlo per altre macchine o applicazioni.

In questo clima, tra il 1952 ed il 1956, iniziarono a circolare espressioni quali *pseudocodice*, *codice automatico*, *programmazione automatica*, con le quali si tendeva a far riferimento a quelle istruzioni che era possibile 'automaticamente' far tradurre alla macchina nella corrispondente forma binaria, permettendo al programmatore di svincolarsi dall'architettura di riferimento.

Quando John Warner Backus (*Riquadro 1*), nella primavera del 1949, prossimo alla laurea in matematica, si trovò a visitare un centro di calcolo dell'IBM nel quale erano in corso le procedure di test di uno dei primi calcolatori elettronici, il Selective Sequence Electronic Calculator¹ (SSEC), si rese immediatamente conto che, se la macchina fosse stata dotata di un sistema in grado di permettere ai programmatori di ridurre il tempo impiegato per l'impostazione di un problema, sarebbe notevolmente migliorata sia la qualità che l'affidabilità del software generato. A seguito della sua proposta, i dirigenti del New York Scientific Computing Service dell'IBM, diedero il via alle sperimentazioni per lo *Speedcoding System* e nel gennaio del 1953 ne affidarono la direzione generale a John Sheldon e la supervisione tecnica proprio a John Backus.

Lo *Speedcoding* fu il nome dato ad un sistema inteso come combinazione integrata di un'ampia gamma di computer digitali ed un metodo mediante il quale questi computer potevano essere programmati per risolvere problemi scientifici e di ingegneria. La famiglia presa come riferimento fu quella del 701², una macchina con memoria a tubi a vuoto³, spesso definita come una «memoria elettrostatica»: per garantire prestazioni veloci ed affidabili, si impose che lavorasse internamente utilizzando il sistema numerico binario, ma coloro che

¹ Lo SSEC non era un calcolatore nel senso moderno del termine: non aveva memoria per l'immagazzinamento dei dati ed i programmi venivano inseriti utilizzando un nastro di carta perforato.

² La prima linea di produzione dell'IBM riguardò il «modello 701» che fu disegnato da Nathaniel Rochester nel 1952 e fu progettato per le esigenze di calcolo del Pentagono, impegnato nella guerra in Corea: per questo motivo fu definito defence calculator. Ne vennero prodotti ed installati solamente diciannove esemplari nell'arco dei suoi tre anni di vita.

³ «Vacuum tube»: dispositivo di amplificazione o di interruzione che opera controllando un fascio di elettroni nel vuoto. Nella forma base, consiste di un filamento costituito da una resistenza riscaldata elettricamente, un catodo collegato al filamento, una griglia ed una piastra. Il filamento produce elettroni che vengono «raccolti» dal catodo ed inviati in direzione della piastra. Se la griglia è negativa rispetto al catodo, gli elettroni vengono respinti e solo un numero molto basso di essi raggiunge la piastra. Se la griglia è positiva, gli elettroni passano attraverso la piastra e la corrente fluisce nel circuito collegato all'uscita. Il tubo funziona come dispositivo di interruzione, con una debole corrente nella griglia che attiva e disattiva una corrente relativamente elevata nella piastra. Se la tensione sulla griglia è contenuta in una banda tale che il suo livello più alto non causa il passaggio di tutta la corrente nella piastra, il tubo funziona come un amplificatore: più alto è il segnale di entrata, più alto è il segnale di uscita.

usavano lo *Speedcoding* non risentivano in alcun modo di ciò poiché tutti i codici venivano introdotti e stampati solo in decimale.

I programmi potevano essere scritti ed immessi nel computer in vari modi, ma solitamente le istruzioni venivano lette su pacchetti di schede perforate. Per istruire la macchina ad eseguire calcoli, apposite unità di registrazione andavano ad incidere su nastri magnetici la sequenza del codice di calcolo, mentre le schede, su cui erano rappresentati dati ed istruzioni, venivano piazzate nel corrispondente lettore. La sola pressione del bottone di avviamento della macchina faceva sì che il programma venisse memorizzato insieme ai dati ed alle istruzioni di inizio della procedura di calcolo, rispettando rigidamente i canoni e le regole imposte dal modello di von Neumann. Da quel momento in poi, le operazioni che il computer eseguiva erano completamente automatiche e tutti i componenti della macchina erano sotto il controllo del programma.

2. Le prime fasi del progetto FORTRAN

Forte dell'esperienza messa a punto con lo *Speedcoding*, Backus fu stimolato a migliorare ulteriormente il sistema di rappresentazione delle istruzioni, cercando di portare la scrittura delle stesse ad un livello di astrazione più alto, un livello espressivo che avrebbe dovuto allontanare il linguaggio dai dettagli minuziosi imposti dalle operazioni svolte dai circuiti digitali, per avvicinarsi alle capacità espressive umane.

L'intuizione di Backus rappresenta probabilmente il punto di svolta nello sviluppo dei linguaggi di programmazione e questo verosimilmente perché, per la prima volta, egli sposta il confine tra l'uomo e la macchina nella direzione dell'essere umano: getta le fondamenta della programmazione ad un livello più alto, livello che andrà crescendo e si perfezionerà con la stessa rapidità con cui si sono percepite le mutazioni tecnologiche delle architetture di calcolo.

Backus presentò al suo manager IBM, Cuthbert C. Hurd, la sua idea di progetto per l'implementazione di un "sistema di programmazione avanzato" che fosse in grado di ovviare alle difficoltà di scrittura di codice a basso livello dell'IBM 704 ed ai problemi di correttezza e di efficienza dei programmi relativi.

Il *Programming System Department* dell'IBM con a capo Cuthbert Hurd, Charles De Carlo e John McPherson, valutarono positivamente le sue proposte tanto che, per i cinque anni successivi al 1953, gli assegnarono i finanziamenti necessari senza domandare mai nulla su come procedeva il lavoro.

Per qualche tempo, a questo progetto non venne dato neanche un nome e alla fine Backus, non sapendo come identificare il linguaggio che stava progettando, prese l'abitudine di riferirsi ad esso come ad un sistema di "traduzione di formule" (*FORmula TRANslating system*). Nel novembre del 1954, il «*Fortran team project*⁴» divulgò un rapporto intitolato *Specifications for the IBM*

⁴ Il gruppo, diretto da Backus, era composto anche da R.J. Beeber, S. Best, R. Goldberg, H.R. Herrick, R.A. Hughes, L.B. Mitchell, R.A. Nelson, D. Sayre, P.B. Sheridan, H. Stern ed I. Ziller. Roy Nutt, dipendente della United Aircraft Corporation, si unì in seguito al gruppo e si dedicò principalmente all'implementazione di tutte le specifiche di input/output del linguaggio. Si veda [22] e [5].

Mathematical FORMula TRANslating System, FORTRAN⁵ (Figura 1) [15], contenente l'idea progettuale di una prima versione del Fortran, identificata con il nome *FORTRAN 0*. Il rapporto evidenziava come i sistemi precedenti offrivano o un codice semplice, a discapito di una esecuzione lenta, oppure una codifica più complessa per esecuzioni veloci. Con il *Fortran* si mirava invece a raggiungere una situazione di compromesso [21, p. 241]:

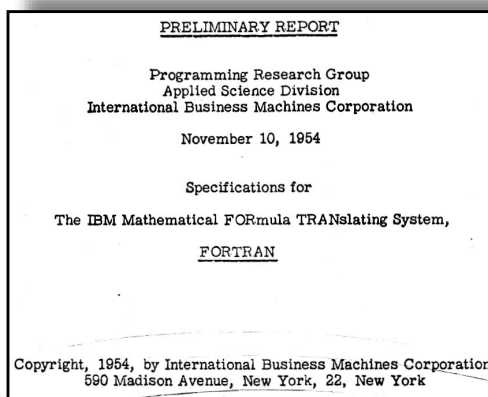


Figura 1
Primo Report sul FORTRAN

«L'IBM Mathematical FORMula TRANslating System, o brevemente *Fortran*, comprenderà un grosso insieme di programmi che abiliteranno l'IBM 704 ad accettare una concisa formulazione di un problema in termini di notazione matematica e a produrre automaticamente un programma veloce per la sua risoluzione.» [21 e 32 p.143]

L'IBM 704 era una macchina dotata di una unità di elaborazione, una unità magnetica centrale di memorizzazione, tamburi e nastri magnetici, un punzonatore ed un lettore di schede, una stampante ed una console composta anche da un monitor⁶. Questo calcolatore ad alta velocità veniva controllato da un programma, residente nella sua memoria interna, composto da istruzioni ad un solo indirizzo. La macchina supportava istruzioni in virgola mobile e consentiva anche l'indicizzazione delle operazioni. Fu disegnata per eseguire elaborazioni rapide, anche con elevate quantità di dati, richieste nella ricerca scientifica, per l'ingegneria, e, in generale, per la produzione di programmi finanziari, industriali e militari. [26] e [6]

Sebbene i tempi per il completamento del progetto furono stimati in circa sei mesi, non ci fu notizia del linguaggio fino a che, nell'ottobre del 1956, venne pubblicato il primo manuale di riferimento [17] nel quale l'IBM precisò che il compilatore sarebbe stato distribuito alla fine dello stesso anno. In realtà, questo impegno non venne mantenuto perché il Dipartimento di Ricerca lo rese disponibile solo nell'aprile del 1957⁷.

⁵ Si noti la sigla IBM nel titolo del rapporto. In effetti il FORTRAN sarà sempre considerato un prodotto IBM e negli anni a venire il successo di IBM andrà di pari passo con quello del FORTRAN.

⁶ Ogni registro di memoria era costituito da un gruppo di 36 nuclei magnetici. Vi erano soltanto due unità di memoria, ognuna delle quali era costituita da 4096 registri. Per ampliare lo spazio disponibile la macchina era stata dotata di memorie ausiliarie costruite mediante dei tubi magnetici. Le «parole», o unità fondamentali di informazione, erano costituite da 36 numeri binari (bit). Per una descrizione più approfondita, si veda [16].

⁷ «Il tempo richiesto e gli sforzi necessari superarono la stima iniziale – situazione comune per chi lavora in questo campo!» [12] e [9]

Il linguaggio non ebbe un immediato successo, al contrario, non destò neppure una minima curiosità. Probabilmente questo era dovuto al fatto che non ne venne mai data una completa ed esplicita definizione, tranne quella informale, che si poté solo desumere dai vari manuali di riferimento. Tuttavia, un articolo scritto da Grace E. Mitchell e pubblicato dall'IBM nel 1957⁸ [24], fu il primo documento che incuriosì la comunità scientifica, tanto che da quel momento ci fu una richiesta sempre crescente nei confronti del linguaggio appena sviluppato. Lo scopo dell'articolo della Mitchell fu quello di illustrare il *Fortran* a scienziati, ingegneri ed a tutti coloro che non erano «specialisti» in materia informatica, fornendo una descrizione «passo passo» di ogni sua singola parte.

Facendo un paragone tra le specifiche preliminari del linguaggio, già pubblicate nel 1954, e quelle evidenziate nel manuale rilasciato nell'ottobre del '56, ci si accorge che effettivamente ci furono pochissime differenze tra una versione e l'altra, sebbene, inevitabilmente, fu comunque praticata qualche piccola variazione⁹. Malgrado fossero stati realizzati, nel corso degli anni successivi al 1954, molti cambiamenti ed aggiunte al linguaggio, e di conseguenza ai relativi manuali, la struttura di base e l'intento dei progettisti furono soltanto ampliati piuttosto che alterati o modificati. William P. Heising dell'IBM Corporation, nel 1963, cioè circa dieci anni dopo la diffusione della prima versione del software, espresse le seguenti importanti valutazioni:

«Il completamento della versione originale del *Fortran* per il 704 rappresenta un significativo successo per la storia della programmazione ed ha indubbiamente avuto influenza per gli sviluppi successivi. Termini comuni quali *programma sorgente*, *programma oggetto* e *macchina oggetto* sono passati nel vocabolario della programmazione grazie all'uso che se ne è fatto nel manuale originale del *Fortran*.» [12]

3. La potenza di un nuovo linguaggio: il FORTRAN I

La prima versione del linguaggio subì alcuni cambiamenti durante il periodo di progettazione ed implementazione, periodo che cominciò nel gennaio del 1955 e continuò sino a che non fu distribuito il compilatore nell'aprile del 1957 [33, p. 45]. Queste variazioni progettuali spinsero il «gruppo» ad identificare la nuova versione con il nome *FORTRAN I*.

⁸ Il testo era suddiviso in tre sezioni, ognuna delle quali descriveva un sottoinsieme del linguaggio, accompagnato da numerosi esempi. La prima edizione apparve nell'autunno del 1957; una edizione rivista e corretta venne pubblicata nel mese di marzo del 1958.

⁹ Le specifiche preliminari del linguaggio indicavano che erano permesse soltanto funzioni con nomi di una o due lettere; nella successiva implementazione risultò invece possibile definire nomi di funzioni che iniziavano con un carattere alfabetico e che potevano essere seguiti da due o più lettere. Oltre a questo, dal rapporto iniziale, venne eliminata la possibilità di mettere insieme numeri ed espressioni (questa opportunità sarà presente soltanto in seguito, con l'ALGOL, ma non ce ne sarà traccia nelle versioni successive del *Fortran*), quella di utilizzare nell'*if* il paragone tra due variabili anziché soltanto un confronto con «zero», e quella di dichiarare esplicitamente il «range» dell'istruzione *do* come una coppia di istruzioni. [32, p. 144].

«[...] fu probabilmente il più complesso linguaggio di programmazione mai prodotto fino a quel tempo ed il fatto che il sistema comprendeva circa 25.000 linee di codice può indicare una misura della sua vastità.» [12]

Per il *FORTRAN I* vennero previste istruzioni di input/output, nomi di variabili di uno o due caratteri, funzioni definite utilizzando tre o più caratteri, espressioni ricorsive, sottoprocedure definibili dal programmatore, formule aritmetiche, l'istruzione di selezione IF e l'istruzione ricorsiva DO. [5, p.71]

L'istruzione logica IF richiedeva l'uso di operatori relazionali del tipo «>» oppure «<», ma il modello 704 non contemplava questi caratteri, per cui gli operatori di questo tipo vennero rimpiazzati utilizzando una «selezione aritmetica» all'interno dell'istruzione IF, così espressa:

IF (espressione aritmetica) N1, N2, N3

dove N1, N2, ed N3 rappresentavano tre istruzioni. Se il valore dell'espressione era negativo, il salto portava all'istruzione etichettata N1; se era zero si saltava ad N2; se era maggiore di zero ad N3. Questa tipologia di rappresentazione è ancora presente nelle versioni più moderne del linguaggio.

Analogamente, per l'istruzione iterativa DO, la sua codifica fu:

DO N1 variabile = primo valore, ultimo valore

dove N1 indicava l'istruzione terminale del ciclo, la «variabile» rappresentava un controllo, mentre «primo valore» e «ultimo valore» (che potevano essere variabili o costanti intere positive) segnalavano, rispettivamente, il parametro iniziale e quello finale del ciclo¹⁰.

Per il *FORTRAN I* non furono previste istruzioni di tipizzazione dei dati: le variabili che avevano nome che cominciava con le lettere I, J, K, L, M ed N furono classificate come variabili di tipo intero, tutte le altre come variabili reali in «virgola mobile». In aggiunta, venne anche incluso un programma di «editing» per consentire eventuali modifiche sul codice. Infine, uno dei nastri magnetici utilizzato per la compilazione venne invece adoperato solo ed esclusivamente come strumento per l'input dei programmi.

La progettazione del linguaggio venne organizzata per sezioni:

«L'intero progetto fu portato avanti grazie ad una stretta collaborazione tra gruppi autonomi e separati costituiti da una, due o, al massimo, tre persone; ogni gruppo era responsabile di una *sezione*: si sviluppavano e concordavano gradualmente le specifiche di input ed output con gli altri gruppi per delineare precisamente il confine tra le varie sezioni.» ([5, p.73])

Nella prima sezione fu prevista una lettura dell'intero codice del programma sorgente e, di conseguenza, la compilazione delle sole istruzioni che potevano

¹⁰ Ad esempio, l'esecuzione dell'istruzione «DO n1 i = k₁, k₂», faceva sì che il gruppo di istruzioni, detto gruppo DO, a partire da quella immediatamente successiva fino a quella contraddistinta da n₁, veniva ripetuto più volte. La prima volta, alla variabile «i», veniva assegnato il valore k₁; successivamente si incrementava «i» di 1 fino a che non si raggiungeva o superava k₂; non appena tale condizione si verificava, il controllo del programma passava all'istruzione immediatamente successiva ad n₁. [33, pp.61-62]

tradursi a questo livello con il successivo salvataggio, in apposite tabelle, di quelle non compilate¹¹. Proprio per questa sua impostazione di compilazione, il *FORTRAN I* venne definito ad «un solo passo», nel senso che il compilatore «vedeva» il programma sorgente solamente una volta. Harlan Herrick fu il responsabile della creazione della maggior parte delle tabelle, Peter Sheridan della compilazione di tutte le espressioni aritmetiche¹² e Roy Nutt della compilazione di tutte le istruzioni di input/output.

Per la seconda sezione si pensò all'analisi delle istruzioni **DO** ed alla generazione del codice corrispondente alle variabili indicizzate, procedendo poi con l'esame delle tabelle prodotte in «sezione 1», anche al fine di ottimizzare gli accessi alla memoria. L'architettura di questa parte del compilatore e tutte le tecniche utilizzate, furono ideate da Robert Nelson ed Irving Ziller i quali progettaron e programmarono l'intera sezione.

La «sezione 3» fu progettata solo ed esclusivamente per convertire l'output delle precedenti nella forma richiesta per le analisi successive: il risultato non fu altro che un *merge* degli output delle sezioni precedenti in un unico programma scritto in codice macchina del 704. Il progetto e la programmazione di questa sezione furono seguiti da Richard Goldberg, un matematico che vi ci lavorò a partire dal novembre del 1955.

Qualche mese dopo, Lois Mitchell Haibt si unì al gruppo del *Fortran* per progettare e programmare la sezione numero 4. In questa fase venne anche contemplata l'analisi statistica della frequenza di esecuzione delle parti del programma sorgente attraverso una sua simulazione. Il programma venne diviso in «blocchi base», blocchi non contenenti istruzioni di salto, e su di esso venne eseguita un'analisi per determinare il percorso e la frequenza di esecuzione degli stessi blocchi (con l'istruzione **FREQUENCY**) per farne un uso più mirato nella sezione successiva¹³. Infine, per minimizzare il tempo necessario per il caricamento e la memorizzazione dei dati e delle istruzioni nei registri di indirizzamento, si pensò di raccogliere informazioni circa il loro uso e di far corrispondere ai registri reali alcuni registri simbolici. [1]

Nella quinta sezione sarebbe poi avvenuta un'ulteriore trasformazione del programma: un codice, che faceva riferimento ad un numero illimitato di registri indice, veniva tradotto in una versione corrispondente facente riferimento ai soli tre registri del 704. Questa sezione fu interamente progettata e programmata da Sheldon Best, che fu «prestato» al gruppo da Charles Adams del Digital Computer Laboratory del MIT. A partire dal 1955, Best disegnò una delle più intricate e complesse sezioni del compilatore ritenuta tra le più solide, innovative

¹¹ Il traduttore creava innanzitutto una sequenza di istruzioni aritmetiche dalle quali venivano eliminati i calcoli ridondanti (infatti il termine «*common subexpression elimination*» apparve di frequente nelle prime documentazioni). [1]

¹² Peter Sheridan ideò ed implementò un numero di trasformazioni ottimali sulle espressioni che vennero utilizzate anche nelle versioni successive del Fortran. [34]

¹³ Per un approfondimento sull'uso di queste tecniche si veda [7].

e significative per le tecniche di gestione degli indirizzi e dei registri di memoria¹⁴.

Nell'ultima sezione, la sesta, si assemblava il programma finale in un programma binario rilocabile; in questa fase veniva prodotta una mappa del programma e dei dati, sunto dell'output in *Fortran*. Roy Nutt, che disegnò e programmò l'intera sezione, scrisse anche tutti i programmi per la gestione dell'I/O.

Dalla primavera del 1956 ai primi mesi del 1957 si effettuarono i test di controllo:

«[...] fu un periodo molto emozionante; spesso rimanevamo attoniti nel verificare ciò che il compilatore eseguiva ed i cambiamenti che portavano alla definizione di un programma oggetto efficiente.» [5, p.74]

Nell'estate del 1956, mentre sembrava vicino il completamento del progetto, si paventò, per la prima volta, un problema che ancora oggi è tenuto in particolare conto: la *stesura della documentazione*. È questo probabilmente l'ulteriore salto di qualità e l'importante tassello che si aggiunge nella direzione della standardizzazione dei linguaggi: la programmazione non è più il semplice mettere insieme istruzioni di codice sorgente ma si identifica con processi più complessi tra cui quello di fornire una documentazione che ne descriva non soltanto i requisiti di sistema ma anche un manuale di utilizzo per l'abbattimento dei tempi di comprensione del software.

Il 15 ottobre del 1956, con una copertina piuttosto grossolana, scritto completamente a mano, con margini molto ampi ed un numero di pagine abbastanza ridotto (soltanto 51¹⁵), fu distribuito il primo *Programmer's Reference Manual* (Figura 2) [17], che rappresentò, per molto tempo, l'unico esempio di manuale di linguaggio di programmazione.

Una delle caratteristiche più rilevanti del *FORTRAN I* non fu descritta nel manuale di riferimento, poiché venne aggiunta al sistema dopo la sua pubblicazione. Questa opzione forniva la possibilità di definire una funzione nel programma sorgente mediante l'uso di una istruzione specifica. Questa istruzione doveva necessariamente precedere il resto del programma e si presentava come un'istruzione di assegnazione che riportava, nella sua parte sinistra, la definizione del nome della funzione e dei suoi argomenti, e, nella parte destra, un'espressione coinvolgente tali argomenti.

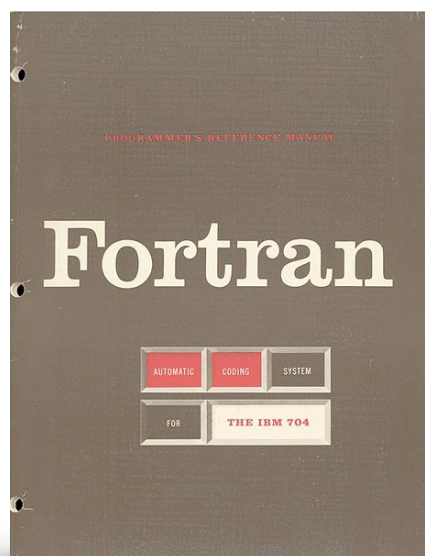


Figura 2
Programmer's Reference Manual
(1956)

¹⁴ Richard Goldberg descrive dettagliatamente questa sezione in un articolo pubblicato nel 1984. La sua familiarità con le tecniche ivi utilizzate ha una spiegazione nel fatto che, insieme a David Sayre, ne esaminò e testò il funzionamento. [10]

¹⁵ La descrizione del linguaggio occupava 21 pagine, 11 erano dedicate alle istruzioni di input/output e le restanti contenevano esempi.

L'IBM commercializzò il 704 corredato del compilatore *Fortran* di cui distribuì una versione aggiornata del suo manuale di riferimento:

«Quando distribuimmo il sistema, consegnammo anche un *Preliminary Operator's Manual* datato 8 aprile 1957. Esso conteneva la lista completa dei simboli dell'intero compilatore più altre informazioni sul sistema e sulla diagnostica¹⁶.» [18]

4. FORTRAN II

L'utilizzo del *FORTRAN I* evidenziò una serie di piccoli errori: fu un periodo difficile, sia per gli utenti che per gli autori, che terminò soltanto quando questi errori, almeno i più frequenti, furono riconosciuti e corretti. Le modifiche fatte sul linguaggio portarono alla stesura della una nuova versione del compilatore *FORTRAN II*, che fu ufficialmente rilasciata dal Programming System Department dell'IBM nel giugno del 1958. [12]

In questa versione del linguaggio vennero coperti molti dei «buchi» della versione precedente¹⁷ ed aggiunte alcune significative funzionalità, la più importante delle quali fu la compilazione indipendente di ogni singola sottoprocedura o funzione. Senza questa possibilità, per ogni cambiamento apportato ad un programma sarebbe stato necessario fare una intera ricompilazione.

Il *FORTRAN II* fu per la maggior parte progettato da Backus, Nelson e Ziller. Mitchell, con la collaborazione di Benyce Brady e LeRoy May, si occupò della scrittura del nuovo codice¹⁸.

Una cosa interessante da segnalare è che né l'IBM, né alcun membro del gruppo che sviluppò questa nuova versione del compilatore, pubblicarono mai un manuale di riferimento ufficiale nel quale si faceva uso esplicito della parola «*Fortran II*». In verità questa versione fu distribuita così velocemente rispetto alla precedente che non si ebbe il tempo di valutarne le migliorie. [30, p.148]

5. FORTRAN III

Durante la fase di sviluppo del *FORTRAN II*, Irving Ziller cominciò a pensare ad un sistema di compilazione più completo ed avanzato rispetto alla versione precedente.

Nella terza versione del *Fortran*, infatti, Ziller aveva previsto la possibilità di scrivere le istruzioni del linguaggio unitamente a quelle del codice macchina

¹⁶ Il manuale era composto da cinque pagine contenenti istruzioni generali e 32 pagine per la gestione degli errori di terminazione di compilazione. Alcuni dei messaggi che il compilatore inviava in caso di errore, citavano: «errore del programma sorgente, arrestare la macchina, correggere la formula in questione e far ripartire l'elaborazione del programma», oppure «parentesi insufficienti o ridondanti nella formula aritmetica o in una formula di tipo IF».

¹⁷ «[...] alcuni difetti progettuali del sistema, che avevamo riscontrato ma che non avevamo avuto il tempo di correggere, furono costantemente sottoposti alla nostra attenzione.» [5, p.76]

¹⁸ Per un approfondimento sulle caratteristiche tecniche del linguaggio *FORTRAN II* si veda [32].

dell'IBM 704 (le istruzioni simboliche del 704 potevano avere variabili *Fortran* come «indirizzi»). La versione «III» permetteva anche l'uso di istruzioni algebriche booleane, il trattamento delle informazioni algebriche, la capacità di passare sottoprogrammi come argomenti e l'inclusione di istruzioni in linguaggio macchina «in linea».

Anche questa *release* fu solamente una versione di “passaggio”: il sistema fu implementato dallo stesso Ziller con la collaborazione di Nelson e Nutt. Non fu mai distribuito ma, per tutto il 1959 ed i primi mesi del '60, se ne resero disponibili solo 20 installazioni per uso interno della società, accompagnate da una breve documentazione. [19] Queste venti versioni divennero operative per gli hardware IBM 709/7090¹⁹, per i quali si utilizzarono solo le istruzioni compatibili con il 704²⁰.

Tra il 1958 ed il 1959 il «progetto *Fortran*» venne affidato all'Applied Programming Department. La responsabilità del progetto non fu più di John Backus, ma di William Heising, direttore della sezione che curava la parte della programmazione dell'IBM 7090. Data la sua ovvia conoscenza del codice della macchina, l'IBM gli affidò anche il progetto relativo allo sviluppo del *Fortran* per quel modello. Seguirono gli anni di transizione tra il *FORTRAN II* ed il *FORTRAN IV*. [14]

6. Gli anni '60: la standardizzazione del linguaggio

Nel 1962 l'IBM pubblicò un bollettino che anticipava la distribuzione di un linguaggio, che prenderà poi il nome di *FORTRAN IV*, «indipendente dalla macchina», un linguaggio, cioè, che «non doveva ricordare il codice macchina».

Tutte le istruzioni proprie del *FORTRAN II* che coinvolgevano, ad esempio, i nastri per l'input/output ed i tamburi di memoria del «704», vennero definitivamente eliminate rendendo il *FORTRAN IV* incompatibile con la sua precedente versione «II».

Le funzionalità offerte da questa nuova versione, furono sostanzialmente legate al tipo di dati che si potevano utilizzare (LOGICAL, DOUBLE PRECISION, COMPLEX, REAL, INTEGER ED EXTERNAL), alle espressioni booleane implementate come argomento dell'operatore IF, alle funzioni e sottoprocedure aventi nomi passati come argomenti di altri sottoprogrammi ed, infine, alle nuove istruzioni DATA e BLOCK DATA.

L'IBM adottò definitivamente il *Fortran* come linguaggio di programmazione per le sue macchine, tanto che venne utilizzata una politica aziendale di vendita, suggerita per la verità da Backus già nel 1954, secondo la quale «ogni calcolatore IBM doveva essere dotato di un sistema simile al *FORTRAN*». [21, 241-242] e [9] Difatti, nel 1958, il compilatore *FORTRAN* venne rilasciato per l'IBM 709 e 650 e, nel 1960, per i modelli 1620 e 7070.

¹⁹ Il modello 709 aveva la stessa architettura dell'IBM 704 ma consentiva di effettuare molte più operazioni di input/output e aveva prestazioni migliori rispetto al modello precedente. L'IBM 7090 fu la versione «transistorizzata» del 709. [8]

²⁰ Ad esempio non fu più possibile utilizzare le istruzioni «in linea» del linguaggio macchina viste le differenze esistenti tra il modello 704 e il 709.

Per evitare che si venisse a creare una certa confusione tra le versioni diverse del *Fortran* e i computer su cui potevano “girare”, l’IBM pubblicò un manuale che includeva un elenco di istruzioni in corrispondenza delle quali veniva specificato il modello di macchina su cui potevano essere implementate. [20]

Furono questi gli anni di maggiore popolarità del *Fortran*: in aggiunta alle versioni «ufficiali», la maggior parte dei produttori di computer e quasi tutte le «software house» cominciarono a produrre la loro personale versione, alcune volte aggiungendovi delle peculiarità non presenti in quelle distribuite dall’IBM²¹ (vedi Riquadro 2).

Nel 1964, nel suo articolo *The Various FORTRANS*, H. Oswald indicò i risultati di una sua ricerca, segnalando l’esistenza di 43 differenti versioni del linguaggio²². L’esistenza di un così elevato numero di «dialetti», rese molto problematico eseguire il trasferimento di programmi da una macchina all’altra, tanto che, nel 1963, l’American Standards Association (ASA, ora conosciuta come American National Standards Association) avvertì la necessità di rilasciare il suo primo «Standard for a Programming Language» a cui venne attribuito semplicemente il nome *FORTRAN*²³.

Nel 1960 l’American Standards Associations, Sectional Committee X3 for Computers and Information Processing, aveva già costituito una sottocommissione, la X3.4 Sectional Subcommittee, per determinare uno standard tra i linguaggi di programmazione comuni. Il 17 maggio del 1962, la X3.4 diede l’avvio ai lavori del gruppo X3.4.3-FORTRAN, per esaminare le proposte riguardanti la standardizzazione del linguaggio *Fortran*.

W.P. Heising ha reso pubblici i documenti ufficiali prodotti dal «Subcommittee» dai quali si desume che:

«L’obiettivo dell’X3.4.3 Working Group dell’ASA è produrre un documento, o una serie di documenti, che definiranno l’ASA Standard per il linguaggio *Fortran*. Il risultante linguaggio standard sarà chiaramente riferito a ciò che era chiamato *Fortran* nel passato. I criteri utilizzati per considerare e valutare gli elementi dei vari linguaggi dovranno comprendere (non in ordine di importanza):

- a. facilità d’uso
- b. compatibilità con le precedenti versioni
- c. estensione dell’applicazione
- d. possibilità di ampliamento
- e. facilità di implementazione corrispondente ad una compilazione ed una esecuzione efficienti.

²¹ Nel 1961 il compilatore fu reso disponibile anche per lo UNIVAC e, nello stesso anno, l’ATLAC, una imitazione del Fortran, fu utilizzata sul Philco 2000. [27]

²² La situazione era tale che gli utenti non erano affatto sicuri che un valore computato su una macchina sarebbe stato lo stesso se prodotto da una computazione su un’altra macchina. [28]

²³ [13], riportato integralmente anche in [32, pp. 154-566].

Il *FORTRAN Standard* faciliterà i trasferimenti da macchina a macchina dei programmi scritti in ASA Standard FORTRAN. Lo Standard servirà come documento di riferimento sia per gli utenti che vorranno raggiungere questo obiettivo, sia per i produttori che programmeranno il prodotto riferendosi ad esso.» [32, pp. 154-566]

L'ASA Working Group, alla fine dei suoi lavori, decise di produrre due standard, il *FORTRAN* e il *Basic FORTRAN*, corrispondenti, rispettivamente, al *FORTRAN IV* e al *FORTRAN II*. Questi standard furono approvati il 7 marzo del 1966 e divennero ufficialmente ASA Standards X3.9-1966 (*FORTRAN* [34]) e X3.10-1966 (*Basic FORTRAN* [36]).

7. Controversie, compromessi e modernizzazioni

L'evoluzione del *Fortran* continuò fino a che, nel 1978, venne rilasciata un'ulteriore versione standardizzata, il *FORTRAN 77* [2], che conservò la maggior parte delle caratteristiche del *FORTRAN IV* con l'aggiunta della gestione delle stringhe di caratteri, delle istruzioni di controllo dei cicli con operatori logici e della clausola opzionale *else* combinata con l'*IF*.

Il *FORTRAN 90* [3] è il nome dato all'ultima versione del *Fortran*. Estremamente differente dalla precedente, questa *release* includeva principalmente operazioni per la gestione degli «array²⁴» e nuove istruzioni di controllo²⁵. Con il *FORTRAN 90* si sviluppò una nuova concezione del linguaggio, secondo la quale da esso potevano essere eliminate tutte quelle istruzioni o caratteristiche ereditate dalle versioni precedenti, ritenute obsolete dal programmatore (ad esempio, la gestione aritmetica dell'*ELSE IF*).

Anche se ci sono stati molti cambiamenti e aggiunte nel corso degli anni, la struttura di base del linguaggio e gli intenti di sviluppo sono stati estesi piuttosto che alterati. Di conseguenza, gli obiettivi che si possono desumere già dal manuale originale relativi alle potenzialità e agli intenti del primo *FORTRAN*, sono chiari e limpidi oggi come sessanta anni fa, quando il primo rapporto fu scritto. Il completamento del linguaggio originale per il 704 costituisce dunque un risultato di estrema importanza nella storia della programmazione e ha indubbiamente influenzato gli sviluppi successivi.

In realtà, è assai difficile comprendere il successo globale del *Fortran* in un momento storico nel quale «il modo» in cui le macchine vengono utilizzate è radicalmente cambiato e gli obiettivi e la logica di programmazione si plasmano e si adeguano ad esigenze completamente diverse. Quello che sicuramente va detto è che gran parte del suo successo è dovuto al suo essere il primo

²⁴ *Allocati e deallocati dinamicamente in memoria mediante operatori appositi: è una svolta radicale dal primo Fortran che gestiva solamente dati statici. [33, pp. 47-48].*

²⁵ *L'istruzione case usata per istruzioni con selezione multipla, exit per uscire dal ciclo forzatamente e cycle per spostare il controllo all'inizio del ciclo.*

linguaggio ad alto livello ampiamente utilizzato²⁶. Esso rappresenta cioè la *lingua franca* del mondo dell'informatica, il linguaggio "di strada" che sopravvive e sopravviverà perché è parte straordinariamente utile di ciò che vi è di "vitale" nella programmazione.

Si possono valutare le grosse potenzialità solo se lo si connette strettamente al suo utilizzo e non propriamente alla sua tecnologia: il contributo più rilevante è sicuramente l'essere stato un linguaggio «usabile» su qualsiasi hardware e aver avuto una grammatica di riferimento molto semplice e di facile comprensione. [11]

Il limite più grande è forse stato quello di essere un linguaggio di programmazione estremamente efficace per la risoluzione di problemi scientifici numerici ma poco adattabile per altri usi: la formulazione di programmi relativi a quesiti di altra natura ne implicava, purtroppo, un uso non appropriato e poco elegante.

Emblematiche sono le parole di Jane Sammet che sintetizzano chiaramente le inutili pecche che si attribuiscono a questo linguaggio:

«[...] la maggior parte degli svantaggi del *Fortran* sono radicati nella sua stessa popolarità: le tante persone che lo adoperano avrebbero piacere ad utilizzarlo per ogni tipo di problema. Quando non soddisfa le loro necessità, ingiustamente, lo criticano.» [32, p.157]

²⁶ Per una completa descrizione dell'evoluzione e dello sviluppo dei linguaggi di programmazione si veda [29].

Riquadro 1 - John Warner Backus

Nato a Philadelphia nel 1924, la sua famiglia benestante gli permise di frequentare la prestigiosa Hill School di Pottstown – Pennsylvania. Non fu un buon allievo e gli anni trascorsi alla Hill School furono contrassegnati da una serie di fallimenti. Ciononostante, riuscì a diplomarsi nel 1942 e ad iscriversi alla University of Virginia, al corso di chimica, solo per seguire la volontà del padre. Anche in questo caso, non fu tra gli studenti più diligenti, al contrario, alla fine del secondo semestre, venne espulso dall'università.



J.W. Backus
1924 - 2007

Nel 1946 si trasferì a New York City dove prese in affitto un piccolo appartamento. Grande estimatore della buona musica, non potendosi permettere di acquistare un sistema Hi-Fi, si iscrisse ad una scuola di tecnico radiofonico per imparare a costruirne uno. Presso questa scuola ebbe l'occasione di aiutare un istruttore ad eseguire i calcoli matematici complessi per la messa a punto di un sistema di amplificazione. In questa occasione si rese conto della sua attitudine e del suo interesse per la matematica e decise di intraprendere questi studi iscrivendosi alla Columbia University di New York.

Nel 1959 ebbe l'idea di introdurre una notazione sintattica formale per la descrizione delle regole grammaticali di un linguaggio di alto livello (Backus-Naur Form: J.W. Backus, *The syntax and semantics of the proposed international algebraic language of Zurich-ACM-GAMM Conference*, Proc. International Conf. Information Processing, UNESCO Paris 1959, pp. 125-32).

Negli anni '70 lavorò alla ricerca di metodi per migliorare la programmazione, sviluppando quella che poi è stata definita la cosiddetta *Programmazione Funzionale*.

Nel 1976 ha ricevuto la medaglia nazionale della scienza e nel 1977 il Premio Turing con la seguente motivazione:

«Per i suoi profondi, autorevoli e durevoli contributi al progetto di sistemi di programmazione ad alto livello, in special modo attraverso il suo lavoro sul FORTRAN e per l'originale e influente proposta di metodi formali per la specifica di linguaggi di programmazione.»

Ha lasciato l'industria dell'informatica nel 1991.

Riquadro 2 - Fortran Implementations (1954 - 1965)

J. A. N. Lee • Pioneer Day

List of FORTRAN implementations

Date	Name	Machine	Author(s)	Location
1954-57	FORTTRAN (04)	IBM 704	Backus et al. ^a	IBM
1957	FORTTRANSIT	IBM 650	Bemer, Alexander, Pessin, Hemmes, May	IBM
1958	650 FORTRAN	IBM 650	Pessin, Wu	IBM
1958	FORTTRAN	IBM 709	"	IBM
1958	FORTTRAN II	IBM 704	Backus et al.; Mitchell, Sheridan, Brady, May	IBM
1958	FORTTRAN III	IBM 704	Ziler, Nelson	IBM
1960	FORTTRAN	IBM 1620	Laffan	IBM
1960	ALTAC	Philco 2000	Rosen, Goldberg	Philco
1960	FORTTRAN	IBM 7070	Alexander et al.	IBM
1960	FORTTRAN	IBM 705	Brittenham, Horning, Kuss, Matheny, Miller, Seldon, Williams	IBM/GUIDE
1960	FORTTRAN	CDC 1604	Cray	CDC
1960	Honeywell Algebraic Compiler	?	Opler	CUC
1961	FORGO	IBM 1620	Davidson, McClure	Wisconsin
1961	FORTTRAN	H-290	Hankins	HIS
1961	GOTRAN	IBM 1620	Laffan, Resta	IBM
1961	Automath-800	H-800	Opler, King, O'Connor, Beeber, Hopkins, Brestwick	CUC
1961	FORTTRAN I	UNIVAC	?	Rem-Rand
1961	FORTTRAN II	LARC	CSC	Rem-Rand
1961	UT FORTRAN	IBM 1620	Lee, Field	Toronto
1961	AFIT FORTRAN	IBM 1620	Pratt	Wright Patterson AFB
1961	1401 FORTRAN	IBM 1401	Haines et al.	IBM
1961	FORTTRAN	B5000	?	Burroughs
1961	FORTTRAN IV	IBM 7090/4	Lerner	IBM
1961	FORTTRAN II	RCA 301	Hux et al.	RCA
1962	FORTTRAN IV	IBM 7040/4	Medlock	IBM
1962	FORTTRAN II	UNIVAC LARC	Erdwinn, Ferguson, Gatt, Malenkoff, Nutt, Richards	CSC
1962	Automath-400	H-400	Boris and Cahill	HIS
1962	FORTTRAN II	RCA 301	Hux et al.	RCA
1963	FORTTRAN IV	Univac 1107 ^a	Anderson, England, Erdwinn, Ferguson, Gatt, Kinney, Martin, Nutt, Perrine	CSC
1963	FORTTRAN IV	UNIVAC III	Balke, Hankins, Nutt, Peterson, van der Woude	CSC
1963	Automath-1800	H-800/1800	Chang	HIS
1963	Automath-1400	H-1400	Roth	CDC
1963	FORTTRAN	CDC 160A	Neuhaus	
1963	FORTTRAN S1	IBM Stretch	Glennie	UKAEA (AWRE)
1963	FORTTRAN IV	IBM 7030	Bishop, Webster	IBM
1963	KINGSTRAN	IBM 1620	Field, Jardine, Lee, Lee, Robinson	Kingston, Ont.
1964	FORTTRAN S2	IBM Stretch	Glennie	UKAEA (AERE)
1964	FORTTRAN IV	Philco 2000	Brown, Rice, Richards, Sidrane	CSC
1964	HARTRAN	Atlas (F)	Pyle	UKAEA (AERE)
1964	FORTTRAN	Orion (F)	Taylor, Harrigan	Rutherford Lab. (UK)
1964	FORTTRAN	SDS910	Dunlap, Ryan	Digtek
1964	FORTTRAN	CDC 3000L	Schumacher	CDC
1964	FORTTRAN	CDC 3600	Tiede	CDC
1964	Fast FORTRAN	CDC 3600	?	Mich. State
1964	FORTTRAN I	RCA 3301	Hux et al.	RCA
1964	FORTTRAN IV	RCA 601	Best	Decision Systems
1965	PUFFT	IBM 7094	Rosen et al.	Purdue
1965	FORTTRAN IV	IBM 7030	MacIntosh, Matheny, McPherson, Peterson, Sexton, Stephens	CSC

10 • Annals of the History of Computing, Volume 6, Number 1, January 1984

Bibliografia

1. Allen F.E., "A Technological Review of the Early FORTRAN Compilers, in "Early Days of FORTRAN", *Annals of the History of Computing*, 6 (1), Jan. 1984, pp. 22-26.
2. ANSI, *American National Standard Programming Language FORTRAN*, ANSI X3.9-1978, American National Standards Institute, New York 1978.
3. ANSI, *American National Standard Programming Language FORTRAN 90*, ANSI X3.198-1992, American National Standards Institute, New York 1992.
4. Backus J.W., "Programming in America in the 1950s", in [23], pp. 130-131.
5. Backus J.W., "The History of Fortran I, II, and III", *IEEE Annals of the History of Computing*, 20(4), Oct.-Dec. 1998, pp.68-78.
6. Bright H., "FORTRAN Comes to Westinghouse-Bettis, 1957", *Annals of the History of Computing*, 1(1), Jul. 1979, p.74.
7. Cocke J., Schwartz J.T., *Programming Languages and Their Compilers*, Courant Institute, NYU, New York 1970, p. 511.
8. Davis G.B., *An Introduction to the IBM 7090/94 Computer*, McGraw-Hill, New York-San Francisco-Toronto-London-Sydney, 1965.
9. Friedman W., "From Babbage to Babel and Beyond: A Brief History of Programming Languages", *Computer Languages*, 17 (1), 1992, p. 7.
10. Goldberg R., "Register Allocation in FORTRAN I", in "Early Days of FORTRAN", *Annals of the History of Computing*, 6 (1), Jan. 1984, pp. 19-20.
11. Greenfield M.N., "The Impact of FORTRAN Standardization", in "Istitutionalization of FORTRAN", *Annals of the History of Computing*, 6 (1), Jan. 1984, p. 33.
12. Heising W.P., "Fortran", *Communications of the ACM*, 6 (3), Mar. 1963, pp. 85-86.
13. Heising W.P., "History and Summary of FORTRAN Standardization Development for the ASA", *Communications of the ACM*, 7 (10), Oct. 1964, p. 590.
14. Heising W.P., "The Emergence of FORTRAN IV from FORTRAN II, in "Istitutionalization of FORTRAN", *Annals of the History of Computing*, 6 (1), Jan. 1984, pp. 31-2.
15. IBM, "Preliminary Report, Specifications for the IBM Mathematical FORMula Translating System, Fortran", *Programming Research Group, Applied Science Division*, IBM Corporation, New York, 10 Nov. 1954, pp. 1-29.
16. IBM, "IBM Electronic Data-Processing Machines – Type 704, Manual of Operation: Preliminary Edition", *International Business Machines Corporation*, 1954.

http://www.cs.virginia.edu/brochure/images/manuals/IBM_704/IBM_704.html
(visto 12 marzo 2014)

17. IBM, *The FORTRAN Automatic Coding System for the IBM 704 EDPM*, Programmer's Reference Manual, IBM Corporation, form n. 32-7026, Oct. 1956.

18. IBM, Preliminary Operator's Manual for the Fortran Automatic coding System for the IBM 704 EDPM, Programming Research Dept., IBM Corporation, New York, 8 Apr. 1957.

19. IBM, *Description of Source Language Additions to the Fortran II System*, Programming Research, Additions to Fortran II, IBM Corporation, New York 1958, (Distributed to users of Fortran III, 12 pp.).

20. IBM, *FORTTRAN (General Information Manual)*, Data Processing Division, IBM Corporation, F28-8074, White Plains, N. Y. 1961, p. 65.

21. Knuth D.E., Pardo L.T., "The Early Development of Programming Languages", in [23], pp.197-273.

22. Lee J.A.N., "Pioneer Day, 1982", *Annals of the History of Computing*, 6 (1), Jan. 1984, pp.8-11.

23. Metropolis N., Howlett J., Rota G.C. (eds.), *A History of Computing in the Twentieth Century*. A collection of essays, Academic Press, New York, 1980.

24. Mitchell G.E., *Programmer's Primer for FORTRAN Automatic Coding System for the IBM 704*, 32-0306-1, IBM Corporation, 1957.

25. Murray Hopper G., "The First Bug", *Annals of the History of Computing*, 3 (3), Jul. 1981, pp.285-286.

26. Nutt R., "Compiler Techniques Available in 1954", in "Early Days of FORTRAN", *Annals of the History of Computing*, 6 (1), Jan. 1984, pp.20-2.

27. Olsen T.M., "Philco/IBM Translation at Problem-Oriented, Symbolic and Binary Levels", *Communications of the ACM*, 8 (12), Dec. 1965, pp. 762-768.

28. Oswald H., "The Various FORTRANS", *Datamation*, 10 (8), Aug. 1964, pp. 25-9.

29. Petrocelli C., *Dalla programmazione automatica alla compilazione : Plankalkul, Fortran, Algol e Lisp*, Pensa multimedia, Lecce, 2003.

30. Rabinowitz I.N., "Report on the Algorithmic Language FORTRAN II", *Communications of the ACM*, 5 (6), Jun. 1962, pp. 327-37.

31. Ridolfi P., Coén H., *Come programmare con il Fortran*, Franco Angeli Editore, Milano, 1980.

32. Sammet J.E., *Programming Languages: History and Fundamentals*, Prentice-Hall series in Automatic Computation, Englewood Cliffs, 1969.

33. Sebesta R.W., *Concepts of Programming Languages*, Addison Wesley, Boston-San Francisco-New York-London-Toronto-Sydney-Tokyo-Singapore-Madrid-Mexico City-Munich-Paris-Cape Town-Hong Kong-Montreal, 2002, 5th ed.

34. Sheridan P.B., "The Arithmetic Translator-Compiler of the IBM Fortran Automatic Coding System", *Communications of the ACM*, 2 (2), Feb. 1959, pp. 9-21.

35. *USA Standard FORTRAN*, United States of American Standards Institute, USAS X3.9-1966, New York, Mar. 1966.

36. *USA Standard Basic FORTRAN*, United States of American Standards Institute, USAS X3.10-1966, New York, Mar. 1966.

Biografia

Carla Petrocelli è docente di Storia e Fondamenti di informatica presso l'Università degli Studi di Bari Aldo Moro. Studiosa di linguistica computazionale e di storia dell'informatica, la sua attività di ricerca si è concentrata in particolare sull'analisi di opere scientifiche del '600 investigate con tecnologie computazionali, e sulla evoluzione dei sistemi di calcolo automatici e dei linguaggi di programmazione ad essi applicati.

E-mail: carla.petrocelli@uniba.it