

L'ELABORAZIONE AUTOMATICA

RIVISTA A CURA DEL CENTRO DI CALCOLO INTERUNIVERSITARIO
DELL' ITALIA NORD-ORIENTALE

VOL. 2 N. 2

10 Novembre 1975

SOMMARIO

Titolo	Autore	pag.
LINGEB - Linguaggio gestione biblioteche .	M. AGOSTI - E. CAOVILLA - M. E. CRESCENTI L. LISSANDRINI - A. RIGONI	1

COMITATO DI REDAZIONE:

Prof. Alessandro Alberigi Quaranta

- » Luciano Biasini
- » Giovanni Castellani
- » Marcello Cresti
- » Giuseppe Evangelisti
- » Giovanni Godoli
- » Giuseppe Mannino
- » Amos Musatti
- » Mario Policastro
- » Renzo Renzi
- » Mario Tanfulla

LINGEB - LINGUAGGIO GESTIONE BIBLIOTECHE

M. AGOSTI - E. CAOVILLA - M. E. CRESCENTI - L. LISSANDRINI - A. RIGONI
(Istituto di Statistica - Università di Padova)

Ricevuto in data 3-4-1975

RIASSUNTO — Il linguaggio definito nel presente lavoro realizza un sistema di ricerca automatica dei documenti che compongono una biblioteca. Tale sistema permette all'utente di giungere in modo rapido al recupero dei documenti a cui è interessato mediante l'uso di predeterminate parole chiave. Gli algoritmi che compongono il sistema sono stati costruiti in modo da rendere possibili i necessari aggiornamenti e sono stati realizzati in linguaggio FORTRAN IV ed implementati sull'elaboratore CDC 6600 del Consorzio Interuniversitario dell'Italia Nord-Orientale.

ABSTRACT — This paper describes the oriented language «LINGEB» for informations storage and retrieval system. A LINGEB language interpreter is written in FORTRAN IV and it was implemented on CDC 6600. This language provides for bringing up to date of stored data base, and also it provides for stored information correction. System peculiarity is easy and quick information retrieval for the user.

PREMESSA.

L'idea da cui ha preso l'avvio il presente lavoro nasce dal desiderio di automatizzare le varie fasi di gestione di una biblioteca. Tale realizzazione prevede un sistema di classificazione automatica mediante il quale catalogare i documenti che compongono la biblioteca ed un sistema di ricerca automatica con il quale giungere al loro recupero.

Relativamente al sistema di classificazione si sono svolti solo degli studi preliminari e si è invece realizzato completamente quello di recupero automatico basato su un criterio di interrogazione mediante parole chiave.

Ciò è stato ottenuto definendo dapprima un linguaggio orientato al problema e, successivamente, costruendo un interprete atto alla sua implementazione.

Sono stati necessari inizialmente degli studi sulla teoria dei linguaggi, sulle strutture informative e sui compilatori.

Ci si è poi rivolti allo studio dei principi di base dell'Information Storage and Retrieval e di quanto la letteratura poteva fornire su sistemi di memorizzazione e ricerca automatica.

Infine si sono effettuate delle ricerche sui fondamenti della classificazione automatica.

Realizzato il linguaggio per la gestione automatica delle biblioteche lo si è applicato alla biblioteca dell'Istituto di Statistica dell'Università di Padova.

Si è cercato, quindi, di valutarne pregi e difetti e di indicarne alcuni suoi futuri sviluppi.

Gli algoritmi utilizzati nel presente lavoro sono stati implementati sull'elaboratore CDC 6600 del Consorzio Interuniversitario dell'Italia Nord-Orientale.

I N D I C E

1. MEMORIZZAZIONE E RECUPERO DELL'INFORMAZIONE.

1.1	Introduzione	pag. 5
1.2	Schema generale di un sistema di I.S.R.	» 6
1.3	Linguaggi di individuazione	» 7
1.3.1	I linguaggi predefiniti	» 7
1.3.1.1	I linguaggi di classificazione gerarchica	» 8
1.3.1.2	I sogettari	» 9
1.3.2	I linguaggi postdefiniti	» 10
1.3.2.1	Il « thesaurus »	» 10
1.3.3	Il linguaggio naturale	» 12
1.4	Criteri di valutazione e di confronto	» 12

2. ELEMENTI DI CLASSIFICAZIONE AUTOMATICA.

2.1	Introduzione	» 14
2.2	Reticoli di partizioni di un insieme finito	» 17
2.3	Misura di similarità	» 21
2.4	Preordinamento associato a una misura di similarità	» 26
2.5	Criterio di classificazione	» 27
2.6	La classificazione applicata alle parole chiave	» 32

3. IL LINGUAGGIO « LINGEB » PER LA GESTIONE DELLE BIBLIOTECHE.

3.1	Introduzione	» 33
3.2	Definizione del linguaggio	» 34
3.3	Semantica	» 36

4. IMPLEMENTAZIONE E NORME PER L'USO DEL LINGUAGGIO LINGEB.

4.1	Introduzione	» 39
4.2	Implementazione del linguaggio LINGEB	» 39
4.2.1	Costruzione del traduttore simultaneo delle parole chiave	» 40
4.2.2	Costruzione della matrice di classificazione	» 40
4.2.3	Costruzione dell'archivio dei libri	» 41
4.2.4	Aggiornamenti	» 41
	Flow-chart	» 42
4.3	Descrizione dei dati di input richiesti	» 51
4.3.1	Input per la costruzione del traduttore simultaneo delle parole chiave	» 51
4.3.2	Input per la costruzione della matrice di classificazione	» 51
4.3.3	Input per la costruzione dell'archivio dei libri	» 51
4.3.4	Input per l'inserimento o per il disinserimento di una parola chiave	» 51
4.3.5	Input per l'aggiornamento di un elemento della matrice di classificazione	» 52

4.3.6	Input per l'inserimento di nuovi libri nel relativo archivio .	pag. 52
4.3.7	Input per la ricerca dell'utente	» 52
4.3.8	Input per la stampa delle informazioni relative ai libri della classe/i individuata	» 53
4.3.9	Input per la fine dell'esecuzione	» 53
4.4	Esempio di applicazione	» 53
	Elenchi delle parole chiave	» 54
	Elenco dei libri	» 66
	Listing del programma	» 76
	Input-output di due ricerche dell'utente	» 101
5.	CONSIDERAZIONI E PROSPETTIVE PER NUOVE RICERCHE.	
5.1	Considerazioni sulla metodologia usata	» 103
5.2	Valutazioni sul sistema realizzato	» 104
5.3	Possibili sviluppi della ricerca	» 104
	BIBLIOGRAFIA	» 106

1. - MEMORIZZAZIONE E RECUPERO DELL'INFORMAZIONE.

1.1 *Introduzione.*

Il continuo sviluppo delle conoscenze umane provoca un inarrestabile aumento di ogni sorta di supporto contenente informazione, per cui diviene sempre più sentito l'interesse verso processi atti a sveltire sia l'immagazzinamento, sia il recupero delle informazioni.

Per fare un esempio di quanto sia necessario, per lo sviluppo della scienza, un rapido e totale recupero di informazioni riguardanti un determinato argomento, si può pensare ad un ricercatore che desidera indirizzare in un determinato settore le sue ricerche. Innanzitutto questo studioso dovrà essere a conoscenza di tutto ciò che è già stato detto sull'argomento, per poter riprendere il discorso da dove è stato interrotto e per non rischiare di giungere a delle conclusioni che sono già state raggiunte. La ricerca di tutto il materiale esistente sull'argomento sarà sempre molto gravosa e lo studioso avrà sempre il dubbio di non essere venuto a conoscenza di certe informazioni esistenti.

Questi problemi sarebbero risolti se si disponesse di un servizio di recupero computerizzato delle informazioni, per mezzo del quale, inoltrata la richiesta, si potrebbero avere in brevissimo tempo le informazioni desiderate.

Per avere un'idea di un altro tipo di problema basta pensare ad una organizzazione composta da un certo numero di persone distinguibili per quanto riguarda, ad esempio, le mansioni svolte o che stanno svolgendo, l'istruzione posseduta, i dati anagrafici, ecc. Se in essa si deve intraprendere un'azione in cui occorrono persone aventi determinate caratteristiche, bisogna vedere dapprima se esiste nell'organico ciò di cui si ha bisogno.

Lo studio della teoria, della tecnica e della pratica che tratta tutte le funzioni e processi necessari per completare il trasferimento dell'informazione prende il nome di « Information Storage and Retrieval » ed in seguito si indicherà, per brevità, con la sigla I.S.R..

L'I.S.R. si trova in un rapporto particolare con il resto della conoscenza, infatti da una parte è uno strumento indispensabile per lo

sviluppo di qualsiasi scienza e dall'altra si basa su concetti ed impostazioni assorbiti dai più disparati rami del sapere.

Si può ritenere, quindi, che tutti i problemi di caratteriologia o di classificazione in senso lato siano legati all'I.S.R.

1.2 Schema generale di un sistema di I.S.R.

Se per documento si intende un qualsiasi tipo di supporto contenente informazioni, cioè articoli, libri, nastri magnetici, ecc., il processo dell'I.S.R. può essere schematizzato come in Fig. 1.1.1. In essa si distinguono due fasi.

Nella prima fase viene esaminato il documento e da esso viene estratto, secondo regole prestabilite più o meno formali, un insieme limitato di dati atti a caratterizzarlo chiamato « termine per l'individuazione », che deve essere tale da appartenere ad un « Linguaggio di individuazione ».

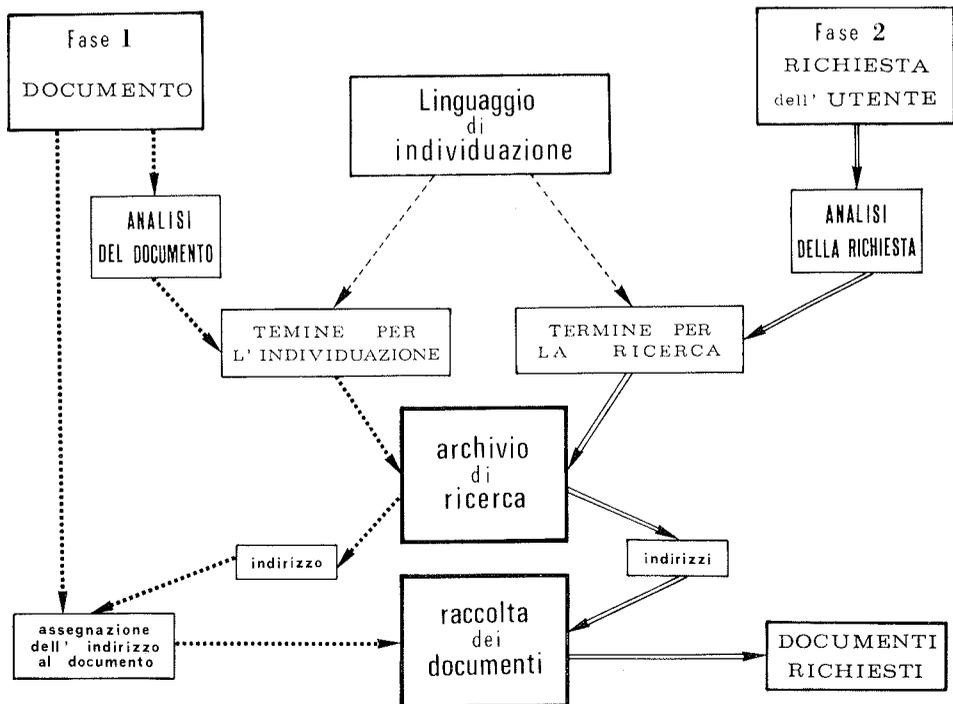


Fig. 1.1.1

Più documenti possono avere lo stesso termine per l'individuazione e cioè appartenere alla classe individuata da quel termine.

Ultimata la prima fase il termine per l'individuazione viene immagazzinato nell'« Archivio di ricerca » mentre il documento viene posto ad un preciso indirizzo nella « Raccolta dei documenti ».

La seconda fase ha lo scopo di recuperare i documenti che costituiscono la raccolta realizzata mediante la fase precedente.

Essa viene attivata dalla richiesta dell'utente, che opportunamente analizzata fornisce il « termine per la ricerca », questo, controllato secondo le istruzioni del linguaggio di individuazione, fa reperire nell'archivio di ricerca l'insieme dei dati che servono per recuperare fisicamente i documenti desiderati.

Per motivi di semplicità ed in vista delle specifiche applicazioni che seguiranno si utilizzerà per l'esemplificazione del problema dell'I.S.R. il caso della gestione delle biblioteche.

1.3 *Linguaggi di individuazione.*

I vari sistemi di I.S.R. si differenziano tra di loro in base al tipo di linguaggio di individuazione che si usa. È però necessario mettere in evidenza che il linguaggio di individuazione viene spesso utilizzato all'interno del sistema di I.S.R. senza essere completamente formalizzato, poiché raramente esso si presta ad esserlo con sufficiente chiarezza e semplicità. Tuttavia l'insieme dei linguaggi di individuazione può essere suddiviso in base a delle caratteristiche fondamentali, per cui si dicono « predefiniti » quelli in cui i termini per l'individuazione vengono completamente stabiliti a priori, mentre « postdefiniti » quelli in cui vengono fissate solo delle regole di composizione dei termini di individuazione.

1.3.1 *I linguaggi predefiniti.*

I linguaggi predefiniti rappresentano il frutto di un lavoro « a priori » eseguito da un gruppo di esperti.

Tali linguaggi di individuazione discendono da un processo di classificazione, cioè di suddivisione in classi, che può essere rivolto all'intero scibile oppure ad una certa disciplina.

I documenti che vengono esaminati sono suddivisi, in base al loro contenuto, in classi e sottoclassi fino al grado di specificità desiderata. Ciascuna classe viene poi contraddistinta con la parola, od il codice,

atta ad indicare la proprietà che gli elementi della classe posseggono in comune. Questa operazione fornisce come risultato un insieme di termini (parole, codici numerici od alfanumerici, brevi frasi) che costituiscono il vocabolario del linguaggio di individuazione e che sono quindi degli « identificatori ».

Quel particolare identificatore costituito da una parola o da una semplice successione di parole che permette di individuare un argomento, cioè che serve come una chiave per introdursi in un particolare contesto, prende il nome di « parola chiave ».

Una volta che si è costituito il linguaggio usando questo metodo elementare di analisi e classificazione, si attribuirà ai documenti la denominazione di una delle classi formata con un processo che è inverso di quello precedente. Si vengono quindi ad usare dei concetti predeterminati. È appunto per questo che linguaggi di individuazione di questo tipo vengono detti predefiniti.

Il vocabolario e la sintassi, cioè l'insieme delle regole che servono per combinare le parole chiave in unità sintattiche più espressive, rappresentano i primi elementi che caratterizzano un linguaggio, pur non essendo la sintassi sempre presente.

Un altro elemento distintivo è costituito dal modo in cui sono organizzati i termini del vocabolario. Si distinguono principalmente due tipi di organizzazione.

Il primo tipo consiste di una lista delle parole chiave organizzata alfabeticamente; l'accesso alla lista è facilitata dalla presenza di un vettore contenente gli indici che permettono di richiamare gli elementi della lista.

Il secondo tipo, invece, è ancora costituito di una lista alfabetica delle parole chiave, in cui però sotto ciascun termine vengono annotate le eventuali relazioni con altri termini della lista.

A seconda del tipo di organizzazione i linguaggi predefiniti prendono il nome di « linguaggi di classificazione gerarchica » e « soggettari ».

1.3.1.1 *I linguaggi di classificazione gerarchica.*

Il più noto linguaggio di classificazione gerarchica è la « Classificazione Decimale Universale » (C.D.U.), che è controllato dalla Confederazione internazionale di documentazione e che riguarda tutto lo scibile umano.

La possibilità di espansione di detto linguaggio è pressoché illimi-

tata, infatti, qualsiasi argomento, per quanto specifico, può essere ulteriormente specificato.

Bisogna qui notare che la relazione di generalità-specificità, implicita tra ciascun termine inferiore e superiore nella gerarchia, rende assai problematiche le variazioni di significato dei codici, sia perché la variazione di significato di un codice si riflette su tutti quelli inferiori, sia perché anche nelle espansioni, e quindi nei codici terminali, si deve tener conto dei significati indotti dal codice di partenza.

Questo costituisce un grosso limite all'uso di linguaggi con tali caratteristiche quando si debba applicare a campi in continua evoluzione.

Un grosso limite è costituito dal fatto che man mano i codici diventano più complessi, e quindi più complessa diventa la loro traduzione in linguaggio naturale, diventa impossibile un indice analitico, a meno che non si considerino come accesso solo i primi numeri della descrizione.

L'indice però perderebbe in questo caso tutto il suo valore di guida per il reperimento del codice appropriato, rendendo l'uso del metodo C.D.U. praticamente possibile solo agli specialisti. Nella sua forma abbreviata, invece, il C.D.U. può risultare estremamente utile sia in fase di classificazione che in fase di ricerca dell'utente, proprio per l'estrema precisione con cui i concetti sono descritti, che esclude qualsiasi ambiguità, e per la facilità di passare da argomenti più generici e più semplici offerta dalla organizzazione gerarchica.

1.3.1.2 *I soggettari.*

I soggettari sono composti di descrittori (soggetti) formulati secondo regole linguistiche che assicurano una certa omogeneità di scrittura.

Sotto ciascun termine del soggettario possono comparire altri termini preceduti dall'indicazione « vedi » (v.) o « vedi anche » (v.a.) per rinviare al soggetto ufficiale o richiamare uno o più soggetti correlati.

Anche in questo caso si possono ripetere alcune considerazioni che sono già state fatte per i linguaggi gerarchici. I soggettari, infatti, fanno uso di un numero fisso di classi, corrispondenti a concetti predefiniti, ma con una struttura assai meno rigida poiché non esistono relazioni gerarchiche e la generica relazione indicata dal « vedi anche » può essere agevolmente modificata senza provocare incongruenze nell'intera struttura del linguaggio.

Questi linguaggi si presentano quindi più flessibili ed aperti all'aggiornamento.

1.3.2 *I linguaggi postdefiniti.*

I linguaggi di questo tipo nascono dalla necessità di trovare una soluzione all'esigenza di avere un accesso multiplo a documenti che trattano una molteplicità di argomenti e di formare inoltre classi di documenti al momento della ricerca, anziché in quello della descrizione.

Gli elementi innovatori che permettono la loro introduzione sono i seguenti:

1) la sostituzione di parole chiave di concetti complessi con parole chiave di concetti elementari (che prendono il nome di unitermini, cioè composti di una singola parola), per cui la formulazione di concetti complessi si rende possibile con una serie di unitermini;

2) l'uso di un sistema multidimensionale per accedere a qualsiasi descrizione da qualsiasi punto di vista.

1.3.2.1 *Il « thesaurus ».*

Le tecniche che si avvalgono dell'uso di linguaggi postdefiniti offrono la possibilità di utilizzare parole chiave unitermini, tratte dal linguaggio proprio dell'autore, onde rimanere fedeli il più possibile alla terminologia originale dei documenti. Però problemi di « falsa coordinazione » da un lato e la varietà del linguaggio naturale dall'altro, possono influire negativamente sia sulla precisione che sulla quantità dei documenti raccolti.

Per ovviare alle difficoltà provocate dalla falsa coordinazione e dalla varietà del linguaggio si possono usare vari strumenti, quali ad esempio l'introduzione di vincoli o legami fra i termini o la specificazione del ruolo che i vari termini giocano. Quello che si rivela essere uno strumento assai utile per raggiungere lo scopo è il « thesaurus », che è un linguaggio controllato, composto di parole chiave unitermini o politermini. Esso è normalmente organizzato in lista alfabetica ed è costruito in modo da rendere esplicite le eventuali relazioni esistenti tra i termini.

Descritto in tal modo il « thesaurus » sembra avere molte analogie con il linguaggio che è stato chiamato « soggettario ». Invece esso si differenzia da quest'ultimo per molteplici aspetti e precisamente:

1) la terminologia del thesaurus riguarda campi limitati dello scibile, ed è per questo meno ambigua;

2) in esso vengono usate molte parole chiave di due o più termini, si riducono in tal modo notevolmente le possibili ambiguità;

3) si introducono le relazioni di « genericità » e « specificità » di un termine rispetto gli altri;

4) i termini che sono correlati semanticamente sono raggruppati in classi e le classi così formate sono organizzate gerarchicamente.

La lista alfabetica è l'indice generale del linguaggio. Essa, infatti, oltre a descrivere le relazioni che ciascun termine ha con altri termini indica:

a) il capostipite della gerarchia nella quale è incluso il descrittore considerato (la mancanza di questa indicazione significa che il capostipite è lo stesso termine considerato);

b) il codice della classe concettuale a cui appartiene la stessa parola chiave.

La lista alfabetica costituisce perciò la via di accesso alle gerarchie e allo schema di classificazione.

Le catene gerarchiche mostrano le relazioni che esistono fra i termini del linguaggio.

La lista classificata suddivide le parole chiave in classi concettuali e stabilisce una gerarchia tra queste classi.

Appartengono alla stessa classe concettuale tutte quelle parole chiave che, essendo fortemente correlate dal punto di vista del significato, possono essere considerate sinonimi nel contesto definito dalla classe concettuale.

Usando il thesaurus come linguaggio di descrizione e di interrogazione è possibile, formulata una domanda con i termini T_1, T_2, \dots :

1) ampliare la raccolta di documenti pertinenti sostituendo a T_i ($i = 1, 2, \dots, n$) le parole chiave che appartengono alla stessa classe di T_i ;

2) rendere più specifica e quindi più precisa la domanda, oppure più generica, riformulandola con parole chiave più specifiche o più generiche;

3) spaziare in tutto il linguaggio, analizzare le relazioni tra parole chiave e decidere, conseguentemente, la più opportuna strategia di ricerca.

Il thesaurus rappresenta il linguaggio tipico dei sistemi conversa-

zionali, infatti per poter essere sfruttato bisogna che vi sia la possibilità, una volta formulata la domanda, di avere una serie di risposte intermedie che permettono di riformularla in modo da avere una risposta soddisfacente per quanto riguarda la quantità dei documenti reperiti ed il grado di precisione di essi.

Il thesaurus può essere registrato nella memoria del calcolatore oppure essere messo a disposizione degli utenti sotto forma di un fascicoletto da usare per istruire il calcolatore sulla strategia di ricerca più opportuna.

Anche il thesaurus, però, può essere affetto da imperfezioni, derivanti da una cattiva costruzione o da un cattivo uso o da ambedue.

Infatti è evidente che un thesaurus non confezionato a misura delle necessità degli utenti oppure che non colga tutti gli aspetti dei singoli documenti non può fornire i risultati desiderati.

1.3.3 *Il linguaggio naturale.*

La pratica sta dimostrando che i linguaggi controllati, anche se adatti a definire concetti specifici, non riescono a raggiungere l'obiettivo di usare la stessa terminologia rappresentativa del linguaggio del documento.

Per questo motivo il linguaggio naturale degli autori tende ad essere sempre maggiormente considerato. Una delle tecniche che ha permesso l'uso del linguaggio naturale è quella delle concordanze.

Altre tecniche, invece, quali l'analisi statistica, l'analisi sintattica, l'attribuzione dei documenti a classi concettuali non sono praticamente usate.

Una applicazione della tecnica delle concordanze è rappresentata dagli indici di parole chiave nel contesto o indici KWIC (Key Words in context) che hanno il pregio di richiedere un procedimento immediato e poco costoso.

1.4 *Criteri di valutazione e di confronto.*

Si sono esaminate in questo capitolo le varie possibilità di impostazione che si offrono a chi decida di organizzare un sistema di I.S.R.. Si presenta ora il problema della scelta fra le diverse impostazioni.

Non ci si dovrebbe porre questo problema se si fosse riusciti a costruire un sistema capace di recuperare tutti e soli quei documenti che interessano i singoli utenti che lo debbono utilizzare.

Giacché non esiste un tale utopistico sistema di I.S.R. si dice che un sistema è tanto migliore o « efficiente » quanto maggiormente s'avvicina a quello utopistico, cioè un sistema è più efficiente di un altro se riesce ad essere più aderente alle richieste dei suoi utenti, essendo entrambi i sistemi considerati, da utilizzare in un certo ambito. Cioè definiti gli scopi che ci si propone di perseguire, si possono stabilire dei criteri atti a valutare in maniera globale un determinato sistema.

Per un qualsiasi sistema di I.S.R. si debbono considerare due quantità: il « coefficiente di richiamo » e il « coefficiente di precisione ».

Se N è il numero dei documenti esistenti in biblioteca, che trattano un determinato argomento e n il numero di quei documenti fra quelli reperiti dal sistema che effettivamente trattano dell'argomento, il coefficiente di richiamo è il seguente rapporto:

$$R = n / N \quad (0 \leq R \leq 1).$$

Si dice che un sistema ha un alto potere di richiamo in relazione a quell'argomento se $R \simeq 1$. Per stabilire un coefficiente di richiamo per tutti gli argomenti trattati nei documenti esistenti nella biblioteca, cioè per stabilire un coefficiente associato al solo sistema, occorrerà calcolare una media, eventualmente ponderata, dei coefficienti di richiamo specifici per argomento.

Si vede subito, però, che basarsi solo su questo coefficiente per valutare un qualsiasi sistema di I.S.R. è insufficiente, infatti l'espressione del potere di richiamo non contiene nessun parametro legato al numero di documenti che si sono richiamati per reperire gli n effettivamente utili.

Occorre allora introdurre la seconda quantità, detta coefficiente di precisione, che tiene conto del numero S di documenti che si sono ottenuti dal sistema in risposta ad una certa richiesta di argomento.

Tale quantità è data dal rapporto:

$$P = n / S$$

dove n è sempre il numero di documenti effettivamente pertinenti ricevuti dal sistema.

Perché un sistema abbia un alto potere di richiamo deve essere $P \simeq 1$.

È evidente che se si tenesse conto solamente di questo secondo coefficiente si perderebbe il confronto con il numero N , effettivamente esistente nella biblioteca, dei documenti che trattano dell'argomento. Infatti, il fatto di avere un alto potere di precisione potrebbe anche voler dire di riuscire a reperire solo una piccola parte dei documenti che trattano l'argomento richiesto e forse neanche i più importanti.

In base a quanto detto si può osservare che i due coefficienti sono spesso dipendenti fra di loro, cioè accade quasi sempre di aumentare il potere di richiamo ma di veder diminuire il potere di precisione e viceversa.

Si possono calcolare, allora, le coppie dei possibili valori dei due coefficienti e scegliere, mediante opportune tecniche di massimizzazione, quella coppia che assicura di ottenere un sistema con una buona efficienza.

Il calcolo dei due coefficienti che permettono, quindi, di valutare un sistema di I.S.R., può apparire molto semplice, invece la misura delle quantità N ed n richiede, per non sfociare in un problema semantico, l'utilizzazione di elaborate tecniche statistiche, quali quelle usate nelle scienze che studiano il comportamento dell'uomo. Ma questo ci può servire come garanzia che i coefficienti di valutazione sono adatti a valutare un sistema di I.S.R. nella sua interezza essendo un sistema di questo tipo sempre strettamente correlato con l'uomo.

2. - ELEMENTI DI CLASSIFICAZIONE AUTOMATICA.

2.1 *Introduzione.*

La classificazione è un'operazione logica che consiste in una distinzione in gruppi o classi di realtà molteplici, secondo la loro somiglianza o differenza. Per quanto essa valga anche nell'ambito strettamente concettuale, ove gli elementi classificati possono non avere una rigida corrispondenza nella realtà esistente, in genere, però, si applica soprattutto nelle scienze sperimentali.

Da un punto di vista generale, l'interesse che la classificazione riveste per le scienze è andato storicamente aumentando man mano che la scienza si è divisa in un gran numero di scienze particolari, che continuamente vanno suddividendosi in rami specializzati. Questa specializzazione è imposta non solo dalla vastità del campo, ma dall'esigenza, per

ogni dominio di studio, di un metodo di ricerca e di un linguaggio appropriato.

La classificazione più ampia costruita dall'uomo è quella relativa al regno animale ed a quello vegetale; sebbene sia contestata nelle sue varie ramificazioni, essa è stata l'indispensabile linguaggio comune a tutti i naturalisti ed ha costituito la base di una elaborazione che suscita ancor oggi numerose ricerche fruttuose.

In tale contesto un'importanza fondamentale assume il concetto di classe « naturale » di cui Beckner diede, nel 1959, una formalizzazione intuitiva.

Una classe di tale tipo è attribuita ad un insieme finito di attributi, tale che ciascun elemento della classe possieda una proporzione importante, ma non specificata, di questi attributi e, reciprocamente, ciascun attributo sia posseduto da un elevato numero di elementi della classe, senza che esso necessariamente sia posseduto da tutti gli elementi della classe.

Una classe così definita viene chiamata « politetica », in opposizione a quella detta « monotetica » definita, nel 1792, di Vicq d'Azyr che definiva una classe per mezzo di una sola proprietà caratteristica, il che si era in seguito dimostrato inadeguato nell'impiego delle scienze naturali ed umane. La definizione data da Beckner è tuttavia legata all'arbitrarietà di un'aspirazione classificatoria in cui lo specialista stabilisce gli attributi caratterizzanti la popolazione in studio con l'unico vincolo di godere della proprietà che il grado di rassomiglianza di due oggetti della popolazione sia elevato se essi appartengono alla stessa classe e piccolo altrimenti.

Un obiettivo pratico immediato è quello di riconoscere un nuovo oggetto appartenente ad una popolazione già studiata, attribuendogli il giusto posto nella classificazione; sapere ad esempio a quale specie appartenga un nuovo esemplare di pesce appena scoperto.

Questo è un problema di ordinamento che non ha nulla a che vedere con quello di classificazione ove lo scopo è quello di definire o formare delle classi.

Nell'ordinamento, in generale, la complessità dei dati osservati è tale da richiedere una moltitudine di calcoli allo specialista perché esso possa prendere una corretta decisione.

Assume pertanto una rilevante importanza il tempo necessario alla operazione sopra accennata, da cui l'interesse sempre più vivo per un « ordinamento automatico » che richieda l'uso di un computer e la costruzione di algoritmi atti a tale scopo.

L'opera di R. Sokal e P. H. A. Sneath « The principles of numerical taxonomy » del 1963 segna un momento decisivo nell'introduzione di metodi automatici.

Il punto di partenza dei metodi proposti sia dagli autori ora citati che da altri che seguirono, è stato la scelta di un indice o misura di similarità, funzione numerica definita sull'insieme dato dal prodotto cartesiano $E \times E$ (*), associando a ciascuna coppia d'oggetti (x, y) con x ed y appartenenti all'insieme E , un numero positivo $S(x, y)$, costruito per misurare la rassomiglianza di due oggetti ed il cui valore viene calcolato facendo ricorso ai due vettori logici:

$$\alpha(x) = (x_1, x_2, \dots, x_i, \dots, x_T)$$

$$\alpha(y) = (y_1, y_2, \dots, y_i, \dots, y_T)$$

ove $x_i = 1$ se l'oggetto x possiede l'attributo α_i e 0 altrimenti, ove $\{\alpha_1, \alpha_2, \dots, \alpha_N\}$ è l'insieme degli attributi. Un discorso analogo vale per le y_i .

A questa fase fa seguito quella relativa alla scelta di un algoritmo capace di stabilire una classificazione od una gerarchia di classificazioni che rispetta le rassomiglianze tra oggetti attraverso i numeri $S(x, y)$. Allora solo l'intuizione dello specialista guidava la scelta della misura di similarità e gli algoritmi proposti erano delle procedure d'approssimazione più o meno giustificate e sovente ambigue, nonostante che i risultati fossero tra loro molto simili.

Era già pertanto viva la preoccupazione di imporre una rigosità matematica nell'elaborazione dei metodi.

Molti autori hanno proposto e ancor oggi propongono metodi tra loro diversi onde poter eliminare quei difetti; dall'algoritmo dei « transferts » definito da Régnier, nel quale tuttavia la imperfezione presente nei dati a disposizione era tale da non poter considerare le differenti partizioni come ugualmente discriminanti, a quello di J. P. Benzecri e W. F. de la Véga che consideravano come dati, non più la similarità $S(x, y)$, ma il sistema di disuguaglianze tra le distanze chiamato ordinamento, che è un ordine totale sull'insieme F della paia di oggetti distinti di E .

(*) La popolazione E di oggetti è in generale un campione di un insieme E' più vasto, che è sovente impossibile osservare nella sua totalità.

Nei prossimi paragrafi, dopo aver introdotto nomenclatura e definizioni utili alla comprensione di quanto verrà detto, vengono descritti dei metodi che presuppongono quale dato di base una informazione relativa alla rassomiglianza degli oggetti.

Questa informazione è in genere stabilita sopra delle apparenze visibili, cioè sulle caratteristiche esterne degli oggetti, senza tener conto dei possibili legami tra questi caratteri, vale a dire della struttura subgiacente che talvolta fa intervenire delle informazioni esterne agli oggetti.

2.2 Reticoli di partizioni di un insieme finito.

In questo paragrafo si fa qualche cenno sul problema riguardante l'insieme delle partizioni di un insieme finito di elementi, e si formalizzano le relazioni intercorrenti fra le varie partizioni.

Sia dato un insieme E di n elementi. Si definisce come partizione di E un insieme finito di classi di E mutuamente disgiunte fra loro, la cui unione dà E .

L'insieme delle partizioni di E viene indicata con $F(E)$.

Costruire una partizione su E , significa definire una « relazione di equivalenza » fra gli elementi dell'insieme.

Infatti dati due elementi di E , indicati con x e y , essi si dicono equivalenti se e solamente se appartengono ad una medesima classe della partizione.

Il grado della relazione di equivalenza P è formalizzato in questo modo:

$$Gr(P) = \{(x, y) \mid x \in E, y \in E \text{ e } x P y\}$$

dove $Gr(P)$ si intende la partizione sotto studio.

Si possono avere diverse partizioni su E e si può così stabilire un ordine nelle partizioni. Se si hanno le due partizioni P e P' si dice che P è più « fine » di P' se

$$P \leq P'$$

e cioè

$$Gr(P) \subset Gr(P')$$

dove $(\forall x, y \in E) : x P y \Rightarrow x P' y$.

Tutto ciò viene ora illustrato con un esempio.

Se si considera l'insieme

$$E = \{a, b, c, d, e, f, g\}$$

e si formano queste partizioni:

$$P = \{\{a, b, c, d\}, \{e, f\}, \{g\}\}$$

$$P' = \{\{a, b, c, d\}, \{e, f, g\}\}$$

esse possono essere rappresentate come nella Fig. 2.2.1.

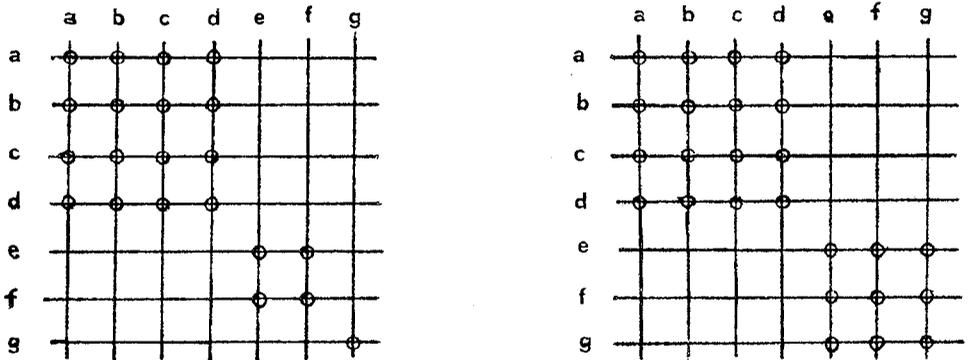


Fig. 2.2.1

L'insieme ordinato $F(E)$ è munito di una struttura a reticolo, cioè ogni coppia di partizioni ha un suo più grande minorante comune che viene indicato con $P \wedge P'$ e un suo più piccolo maggiorante comune indicato con $P \vee P'$.

Il più grande minorante comune è definito per mezzo di un grafo come $Gr(P \wedge P') = Gr(P) \cap Gr(P')$, e cioè $x(P \wedge P')y$ se e solamente se xPy e $xP'y$.

Il più piccolo maggiorante comune invece è definito su uno grafo come $Gr(P \vee P') = Gr(P) \cup Gr(P')$ dove $Gr(P \vee P')$ è il più piccolo grafo della relazione d'equivalenza contenente l'insieme $Gr(P) \cup Gr(P')$.

Esempio:

$$P = \{\{a, b, c, d\}, \{e, f\}, \{g\}\}$$

$$P' = \{\{a, b\}, \{c, d\}, \{e, f, g\}\}$$

$$P \wedge P' = \{\{a, b\}, \{c, d\}, \{e, f\}, \{g\}\}$$

$$P \vee P' = \{\{a, b, c, d\}, \{e, f, g\}\}.$$

L'insieme delle partizioni $F(E)$ dipende esclusivamente dall'intero n , cardinale di E . Nella struttura del reticolo vi sono due estremi:

1) l'estremo superiore che è la partizione più fine in cui ad ogni classe appartiene un solo elemento;

2) l'estremo inferiore che è la partizione più grossolana in cui vi è una sola classe.

Si fanno ora alcuni cenni sulle proprietà del reticolo F .

A) Una partizione P' copre una partizione P se si ha:

$$1) P \leq P'$$

$$2) \{Q \in F \mid P < Q < P'\} = \emptyset$$

e cioè non esiste una partizione Q che stia strettamente fra P e P' .
 P' si può ottenere facilmente riunendo due classi di P .

B) F è un reticolo atomico, dove gli atomi di F sono gli elementi di F che coprono la partizione più fine.

C) Una « catena » del reticolo F è un sottoinsieme totalmente ordinato di F , che si indica con

$$C = (P_1, P_2, \dots, P_k)$$

dove

$$P_1 < P_2 < \dots < P_k \quad \text{con } P_i \in F.$$

La lunghezza di una catena è uguale al numero di elementi che essa contiene meno uno.

Disponendo ora della definizione di catena di una partizione, si possono dare due nozioni fondamentali per un reticolo:

1) l'altezza di una partizione P , che si indica con $h(P)$, è la massima delle lunghezze della catena che si possono formare dall'estremo superiore alla partizione P .

Si osserva ora che se si indica con $c(P)$ il numero delle classi contenute nella partizione $P \in F$, l'altezza viene data dalla seguente operazione:

$$h(P) = n - c(P);$$

2) l'altezza del reticolo F è data dalla lunghezza della catena che unisce l'estremo superiore a quello inferiore, che essendo il meno fine contiene una sola classe.

Per questo motivo si ha:

$$h(P) = n - 1.$$

D) F è un reticolo graduato dall'altezza h , e cioè:

$$1) P' > P \Rightarrow h(P') > h(P);$$

$$2) \text{ se } P' \text{ copre } P, \text{ allora } h(P') = h(P) + 1.$$

Ora si dà un esempio in Fig. 2.2.2 di reticolo sull'insieme $\{a, b, c, d\}$.

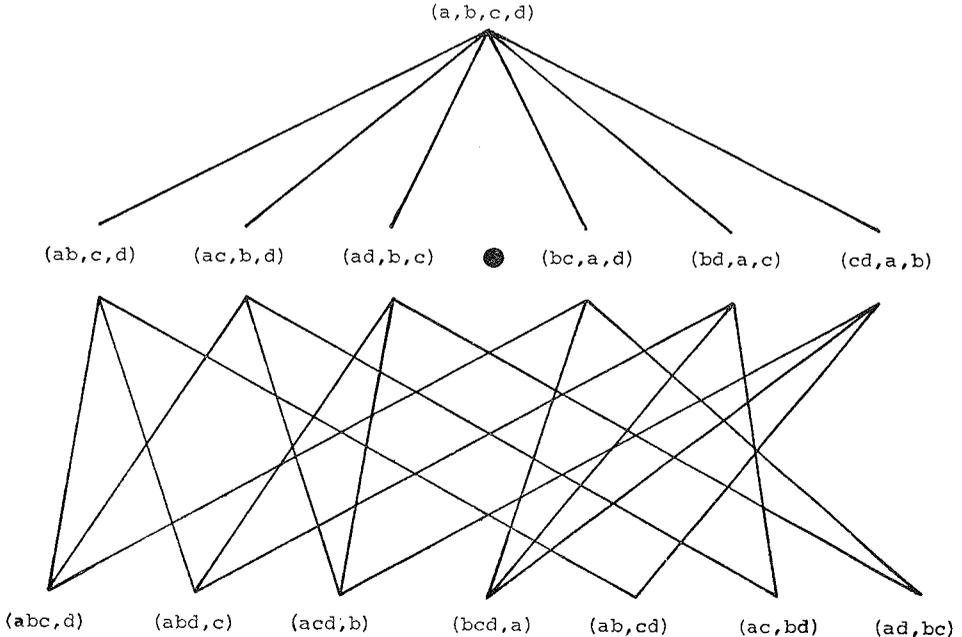


Fig. 2.2.2

2.3 Misura di similarità.

Dopo aver descritto tutte le possibili partizioni che si possono formare su un insieme finito di elementi, e le proprietà che le legano, sorge ora la necessità di costruire delle classi i cui elementi siano raggruppati in maniera soddisfacente rispetto a un criterio di classificazione.

A questo scopo si suppone di stabilire un insieme finito di caratteri che si indica con $\{a_1, a_2, \dots, a_i, \dots, a_p\}$ sufficiente a descrivere tutti gli elementi di E . Ad ogni carattere a_i è associato un insieme di diverse modalità (o valori) che si indica con $A_i = \{a_i^1, \dots, a_i^i, \dots, a_i^k\}$.

Si può, relativamente alla descrizione degli oggetti di E , restringere lo studio solo a qualche modalità. L'assegnazione di una modalità a_i^i ad un oggetto $x \in E$ viene rappresentata a modo di variabile logica che vale 1 se x possiede a_i^i o 0 se non la possiede

A questo punto è bene rimarcare che si usa il termine « attributo » per definire ogni modalità di un carattere necessaria alla descrizione di E .

Si definisce allora il seguente insieme degli attributi:

$$A = \{\alpha_1, \alpha_2, \dots, \alpha_T\}.$$

Se si considera un generico elemento x di E , esso possiede determinati attributi. Il sottoinsieme degli attributi posseduti da x viene indicato con X , ($X \subset A$), e l'elemento x viene rappresentato in $\{0, 1\}^T$ per il punto $\alpha(x) = (x_1, x_2, \dots, x_i, \dots, x_T)$ dove x_i è la variabile logica.

Tutti questi dati vengono generalmente raccolti in una tabella, detta « Matrice d'incidenza », dove gli elementi delle righe rappresentano gli attributi e le colonne gli elementi di E .

Si prendono in considerazione due elementi qualsiasi di E , x e y . Essi sono definiti nel modo seguente:

$$\alpha(x) = (x_1, x_2, \dots, x_i, \dots, x_T)$$

$$\alpha(y) = (y_1, y_2, \dots, y_i, \dots, y_T)$$

dove x_i e y_i possono assumere solo i valori $\{0, 1\}$.

Esaminando il valore assunto da x_i e y_i per un medesimo attributo α_i , si possono trovare diverse combinazioni dei valori $\{0, 1\}$. Più specificamente si ha « associazione positiva » quando l'attributo α_i è presente simultaneamente nei due elementi x e y e cioè $x_i \cdot y_i = 1$, mentre vi è « associazione negativa » quando l'attributo è assente in entrambi.

A queste due combinazioni si aggiungono i casi in cui l'attributo α_i sia posseduto da x e non da y , oppure il caso contrario.

È da notare che il contributo portato alla misura di rassomiglianza di due oggetti dalle associazioni positive è uguale a quello delle associazioni negative.

Esaminate ora le diverse combinazioni, si passa a descrivere il metodo di sintetizzazione dei vari tipi di associazioni. Questa operazione viene eseguita assegnando alla coppia di elementi sotto studio i seguenti numeri cardinali:

s = numero delle associazioni positive

t = numero delle associazioni negative

u = numero degli attributi posseduti da x e non posseduti da y

v = numero degli attributi posseduti da y e non posseduti da x .

Se si indicano con X e Y i sottoinsiemi degli attributi posseduti rispettivamente dagli elementi x e y si può affermare:

$$s = |X \cap Y| = \sum_{i=1}^T x_i \cdot y_i$$

$$t = |X^c \cap Y^c| = \sum_{i=1}^T (1 - x_i)(1 - y_i)$$

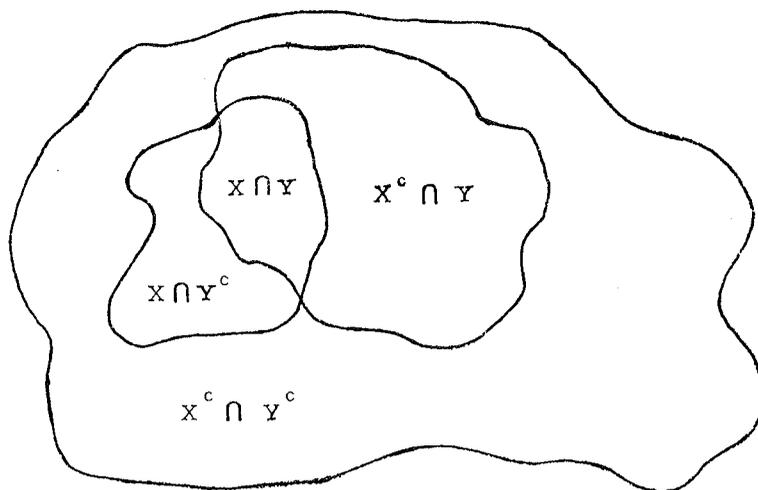
$$u = |X \cap Y^c| = \sum_{i=1}^T x_i(1 - y_i)$$

$$v = |X^c \cap Y| = \sum_{i=1}^T (1 - x_i)y_i$$

dove X^c e Y^c sono i complementi di X e Y sull'insieme A .

Si nota come $s + u = |X|$ e $s + v = |Y|$ e i quattro numeri cardinali siano legati fra loro con questa espressione:

$$s + t + u + v = T.$$



Dei quattro numeri cardinali sopra esposti si considera la tripla $\{s, u, v\}$ che è l'indicatore di similarità dei due elementi (x, y) . L'indicatore di similarità è scritto con questa notazione:

$$I(x, y) = (s, u, v)$$

che è il dato di base indicante la rassomiglianza tra x e y .

Nella Tab. 2.3.1 si dà un esempio della matrice d'incidenza su cui si calcola l'indice $I(x, y)$ che viene riportato nella Tab. 2.3.2.

Dopo aver mostrato il procedimento che porta al calcolo delle misure di similarità se ne dà ora la sua definizione: una misura di similarità è una funzione reale positiva U definita nell'insieme $E \times E$ che si presenta sotto la forma:

$$(x, y) \rightarrow U(x, y) = S(I(x, y)) = S(s, u, v)$$

dove $S(s, u, v)$ è una funzione definita sul sottoinsieme $N^3: \{(s, u, v); s+u+v \leq T\}$, crescente per rapporto a s , simmetrico in u e v e decrescente rispetto a u , dove la crescita per rapporto a s o la decrescita per rapporto a u è stretta.

Questa definizione di similarità è stata introdotta da Lerman ed è quella che verrà utilizzata nel seguito, pur avendo altri autori dato delle definizioni che si differenziano da questa.

	a	b	c	d	e	f	g
α_1	1	0	1	0	0	1	0
α_2	0	1	0	1	1	0	1
α_3	1	1	1	0	0	0	0
α_4	1	0	1	0	1	0	1
α_5	0	1	0	0	0	1	0
α_6	1	0	0	1	0	0	1
α_7	0	0	0	1	0	1	0
α_8	0	1	0	1	1	1	0
α_9	0	0	1	0	0	1	1
α_{10}	1	0	0	0	1	0	1
α_{11}	0	1	0	0	0	1	0
α_{12}	0	1	0	0	1	0	0
α_{13}	1	0	1	1	0	1	1
α_{14}	1	0	0	1	1	0	0

Tab. 2.3.1

	a	b	c	d	e	f	g
a	(7,0,0) 0	(1,6,5) 11	(4,3,1) 4	(3,4,3) 7	(3,4,3) 7	(2,5,5) 10	(4,3,2) 5
b		(6,0,0) 0	(1,5,4) 9	(2,4,4) 8	(3,3,3) 6	(3,3,4) 7	(1,5,5) 10
c			(5,0,0) 0	(1,4,5) 9	(1,4,5) 9	(3,2,4) 6	(3,2,3) 5
d				(6,0,0) 0	(3,3,3) 6	(3,3,4) 7	(3,3,3) 6
e					(6,0,0) 0	(1,5,6) 11	(3,3,3) 6
f						(7,0,0) 0	(2,5,4) 9
g							(6,0,0) 0

Tab. 2.3.2

2.4 Preordinamento associato a una misura di similarità.

Scelto un determinato tipo di indice di similarità, si forma, sulle coppie di elementi dell'insieme E , un preordine totale F che può essere formalizzato nel modo seguente:

$$(\{x, y\}, \{z, t\}) \in F \times F : \{x, y\} \leq \{z, t\} \Leftrightarrow S(x, y) \geq S(z, t).$$

Il preordinamento associato alla misura di similarità S sull'insieme E può essere notato con $u(S)$, che è il risultato dell'applicazione di S su E .

Qui ci si serve di misure di similarità del tipo: $s; u+v; u \cdot v$. Questi diversi criteri portano a preordinamenti distinti che, in caso di scelta, possono essere confrontati fra loro. Qui di seguito si danno degli esempi riferiti alla Tab. 2.3.1 usando le misure di similarità $s, u+v$.

Il preordinamento rispetto a s è il seguente (*):

$$\begin{aligned} \{a, c\} = \{a, g\} < \{a, d\} = \{a, e\} = \{b, e\} = \{b, f\} = \{c, f\} = \{c, g\} = \\ = \{d, e\} = \{d, f\} = \{d, g\} = \{e, g\} < \{a, f\} = \{b, d\} = \{f, g\} < \{a, b\} = \\ = \{b, c\} = \{b, g\} = \{c, d\} = \{c, e\} = \{e, f\}. \end{aligned}$$

Il preordinamento rispetto a $(u+v)$ è, invece, il seguente:

$$\begin{aligned} \{a, c\} < \{a, g\} = \{c, g\} < \{b, e\} = \{c, f\} = \{d, e\} = \{d, g\} = \{e, g\} < \\ < \{a, d\} = \{a, e\} = \{b, f\} = \{d, f\} < \{b, d\} < \{b, c\} = \{c, d\} = \{c, e\} = \\ = \{f, g\} < \{a, f\} = \{b, g\} < \{a, b\} = \{e, f\}. \end{aligned}$$

In questi due esempi i valori assunti da s sono nell'ordine: 4, 3, 2, 1, mentre quelli di $u+v$ sono: 4, 5, 6, 7, 8, 9, 10, 11.

Da quanto esposto precedentemente si possono trarre due importanti conclusioni:

1) due misure di similarità sono equivalenti su un insieme E dato, se e solamente se, i preordinamenti rispettivamente associati su E sono identici. Perché si verifichi tutto questo si deve trovare una funzione f tale che:

$$S'(x, y) = f[S(x, y)]$$

dove con S e S' si sono indicate le due diverse misure di similarità.

(*) In questo contesto si usa la notazione « < » con il significato di separatore fra coppie di diversa misura di similarità.

2) Se il numero d'attributi posseduti per un medesimo oggetto di E è invariabile, tutte le similarità sono equivalenti su E .

Questa affermazione può essere formalizzata nel modo seguente:

$$I(x, y) = (s, l - s, l - s)$$

dove:

$$s = \sum_{i=1}^T x_i \cdot y_i$$

l = numero d'attributi comune a tutti gli elementi di S .

Tutto questo può essere messo sotto la forma:

$$S(s, l - s, l - s) = R(s)$$

dove $R(s)$ è una funzione numerica a valori positivi definita su $\{0, 1, 2, \dots, T\}$ e strettamente crescente.

2.5 Criterio di classificazione.

Si è definito precedentemente, per ogni coppia di elementi di E , l'indice di similarità S e il preordinamento associato. A questo punto si cerca, sull'insieme E , di definire una partizione e una catena del reticolo delle partizioni che rispetti in maniera soddisfacente la rassomiglianza fra gli oggetti. Per fare ciò si deve definire un criterio di base, ove per criterio di base si intende generalmente una relazione di preordinamento (che può anche essere parziale) sull'insieme delle classi. Per tale relazione, C è preferibile a C' se C è più vicino di C' alla struttura definita nel dato di base. I molti criteri esistenti al riguardo partono da dati di base diversi, in questa esposizione si tiene come partenza la « matrice d'incidenza », vista precedentemente. Il dato di base è allora una matrice i cui elementi sono indicati con z_{ij} dove:

$i = 1, 2, \dots, T$ che sono gli attributi

$j = 1, 2, \dots, n$ che sono gli elementi di E .

Con questa suddivisione ogni singolo elemento di E è rappresentato da una colonna della matrice, e una partizione di E è formata da una partizione di colonne. Per comodità è utile unire le colonne che appartengono ad una medesima classe.

Sia $(E_1, E_2, \dots, E_i, \dots, E_k)$ una partizione di E in k classi. Se si considera la classe E_i si definisce un sottoinsieme A_i degli attributi tale che ogni carattere di A_i sia frequente su E_i .

È bene precisare che il sottoinsieme A_i non forma necessariamente una partizione ma generalmente costituisce un ricoprimento, poiché un medesimo attributo può intervenire nella formazione di più classi. Se il ricoprimento è abbastanza debole, si può affermare che le parti A_i sono facilmente estensibili. Infatti se si considera il rapporto:

$$\frac{|A_i \cap A_j|}{|A_i \cup A_j|}$$

ed esso assume un valore elevato, si può affermare che non vi alcuna ragione per formare due classi distinte.

Da tutto ciò segue che lo scopo del criterio di classificazione è quello di trovare una coppia dove il primo argomento è una partizione su E e il secondo una famiglia di parti di A , ricoprente A , tale che gli elementi della famiglia e le classi della partizione si corrispondono biunivocamente. Una tale coppia può essere formalizzata nel modo seguente:

$$(E_i, A_i \mid i \in I).$$

Data la coppia $(E_i, A_i \mid i \in I)$ si può costruire una nuova matrice di zero e uno. Infatti, se si indica con L la restrizione della matrice d'incidenza alla regione $\bigcup_{i \in I} A_i \times E_i$ e con M il complementare di L , si può formalizzare la nuova tabella, indicata con B , nel seguente modo:

$$\begin{aligned} \bar{z}_{ij} &= 1 && \text{se } (i, j) \in L \\ &= 0 && \text{se } (i, j) \in M = L^c. \end{aligned}$$

Si vede come una tale matrice rappresenti il caso ideale dove la classificazione $(E_i)_{i \in I}$ si impone in modo evidente. Da ciò si deduce che un oggetto appartiene alla classe E_i se e solamente se esso possiede esattamente il sottoinsieme d'attributi A_i .

Si danno, ora, due tipi di criteri, dove ognuno parte da una forma di restrizione del problema.

Le due impostazioni sono dovute alla diversa ottica con cui è vista l'assegnazione di un attributo ad una classe A_i .

1) Nel primo caso si impone che la famiglia A_i sia una partizione. Con questa assunzione si corre il rischio d'assegnare ad una sola classe un attributo presente in modo consistente anche in altre classi. Malgrado questo inconveniente si può sperare di assegnare ad una classe gli attributi che più la specificano. L'assegnazione dell'attributo alla classe viene eseguita nel modo seguente:

data la partizione E_i , si calcola l'espressione:

$$\sum_{i, j} (z_{ij} - \bar{z}_{ij})^2$$

e si assegna l'attributo alla classe che ha il valore meno elevato.

2) Nel secondo caso invece si assegna l'attributo dato alle classi dove compare più frequentemente, cioè in senso largo rispetto al primo caso. Pertanto si calcola per ognuna delle classi il rapporto:

$$\frac{n_k^i}{n_k}$$

dove n_k è il cardinale della classe e n_k^i è il numero di volte che l'attributo è presente nella k -ma classe.

L'attributo si assegna alle classi che hanno il rapporto più elevato.

Costruite le classi degli attributi si sono trovate varie coppie del tipo $(E_i, A_i | i \in I)$.

A questo punto è possibile definire il criterio. Fra due coppie si preferirà quella in cui il valore di

$$\sum_{i, j} (z_{ij} - \bar{z}_{ij})^2$$

è il meno elevato dove $B = \|\bar{z}_{ij}\|$ è la matrice di zero e uno relativa a $(E_i, A_i | i \in I)$.

Ora si riporta un esempio dove si confrontano le due partizioni:

$$\{\{a, c, g\}; \{b, d, e, f\}\} \quad (\text{I})$$

$$\{\{a, c, g, b\}; \{b, e, f\}\} \quad (\text{II})$$

Per rendere più semplice il confronto si è riscritta in Tab. 2.5.1 la

	a	c	g	b	d	e	f
α_1	1	1	0	0	0	0	1
α_2	0	0	1	1	1	1	0
α_3	1	1	0	1	0	0	0
α_4	1	1	1	0	0	1	0
α_5	0	0	0	1	0	0	1
α_6	1	0	1	0	1	0	0
α_7	0	0	0	0	1	0	1
α_8	0	0	0	1	1	1	1
α_9	0	1	1	0	0	0	1
α_{10}	1	0	1	0	0	1	0
α_{11}	0	0	0	1	0	0	1
α_{12}	0	0	0	1	0	1	0
α_{13}	1	1	1	0	1	0	1
α_{14}	1	0	0	0	1	1	0

Fig. 2.5.1

matrice d'incidenza precedentemente vista, riordinando in modo opportuno le colonne.

Con riferimento alla partizione (I) si ottiene:

In base al primo criterio di associazione degli attributi:

$$(A_1, A_2) = (\{\alpha_1, \alpha_3, \alpha_4, \alpha_6, \alpha_9, \alpha_{10}, \alpha_{13}\}, \\ \{\alpha_2, \alpha_5, \alpha_7, \alpha_8, \alpha_{11}, \alpha_{12}, \alpha_{14}\}) \\ \sum_{i,j} (z_{ij} - \bar{z}_{ij})^2 = 26.$$

In base al secondo criterio di associazione degli attributi:

$$(A_1', A_2') = (\{\alpha_1, \alpha_3, \alpha_4, \alpha_6, \alpha_9, \alpha_{10}, \alpha_{13}\}, \\ \{\alpha_2, \alpha_5, \alpha_7, \alpha_8, \alpha_{11}, \alpha_{12}, \alpha_{14}\}) \\ \sum_{i,j} (z_{ij} - \bar{z}_{ij})^2 = 26.$$

Con riferimento invece alla partizione (II) si ottiene:

In base al primo criterio di associazione degli attributi:

$$(A_1, A_2) = (\{\alpha_1, \alpha_3, \alpha_4, \alpha_6, \alpha_9, \alpha_{10}, \alpha_{13}\}, \\ \{\alpha_2, \alpha_5, \alpha_7, \alpha_8, \alpha_{11}, \alpha_{12}, \alpha_{14}\}) \\ \sum_{i,j} (z_{ij} - \bar{z}_{ij})^2 = 36.$$

In base al secondo criterio di associazione degli attributi:

$$(A_1', A_2') = (\{\alpha_1, \alpha_3, \alpha_4, \alpha_6, \alpha_9, \alpha_{10}, \alpha_{13}\}, \\ \{\alpha_2, \alpha_5, \alpha_7, \alpha_8, \alpha_{11}, \alpha_{12}, \alpha_{14}\}) \\ \sum_{i,j} (z_{ij} - \bar{z}_{ij})^2 = 34.$$

Da questo esempio si può dedurre come la migliore partizione su E sia $\{\{a, c, g\}, \{b, d, e, f\}\}$.

2.6 La classificazione applicata alle parole chiave.

Dopo aver illustrato la classificazione di un insieme d'oggetti in classi, si vede ora l'applicazione di questa teoria alle parole chiave.

La classificazione si presenta più complessa per il fatto che una parola chiave può assumere vari significati e può avere diverse connessioni con altre parole chiave.

Tutto questo è stato formalizzato con quattro tipi diversi di classificazione a cui possono essere ricondotte le varie connessioni esistenti fra le parole chiave. È utile dire innanzitutto che essi sono caratterizzati da una sofisticazione crescente, così che la classe di tipo 1 è la più semplice e quella di tipo 4 la più complessa. Questa è un'osservazione sia riguardo la loro natura logica e sia la procedura di ricerca che è tipicamente associata ad essi.

Il modo migliore per introdurli è quello di illustrarli, dando un esempio di una classe di ogni tipo, come nella Fig. 2.6.1.

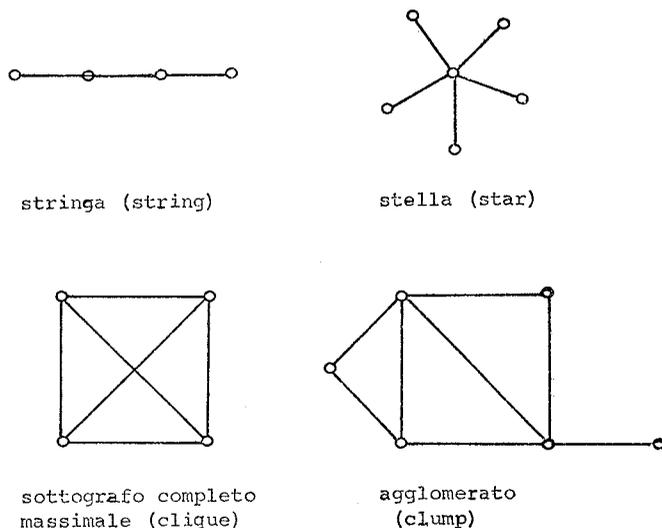


Fig. 2.6.1

Nella Fig. 2.6.1 i nodi sono le parole chiave, i collegamenti sono le connessioni di similarità fra gli elementi della classe che li tengono assieme, cioè le connessioni interne sulle quali dipende l'esistenza della classe.

Esaminando i vari tipi di classificazione, la stringa può essere considerata come la forma più elementare di una classe: si prenda uno specifico elemento, ad esso si trova connesso un altro elemento, e poi un altro elemento connesso con quest'ultimo e così via.

Nella stella invece si prende un singolo elemento e si seleziona una serie di elementi connessi con esso.

Il terzo tipo, il sottografo completo massimale, si ottiene identificando un insieme di elementi che sono tutti connessi con ogni altro. Il tipo più sofisticato, l'agglomerato, dipende da una grande densità di connessioni sia nell'insieme che fuori di esso; il diagramma dell'agglomerato è quindi piuttosto ingannevole perché sono omesse le connessioni esterne complementari.

Osservando la Fig. 2.6.1 si può notare come la stringa e la stella siano basate su elementi individuali, mentre il sottografo completo massimale e l'agglomerato siano basati su insiemi di elementi. Inoltre la stringa, la stella e il sottografo massimale impongono condizioni assolute di connessione sugli elementi della classe, mentre l'agglomerato è definito relativamente.

Questi quattro diversi tipi di classificazione possono essere utilizzati per rappresentare tutte le varie situazioni di connessioni esistenti fra le classi di parole chiave.

3. - IL LINGUAGGIO « *LINGEB* » PER LA GESTIONE AUTOMATICA DELLE BIBLIOTECHE.

3.1 *Introduzione.*

Nel seguente capitolo si dà la formalizzazione del linguaggio LINGEB e la spiegazione semantica delle frasi che non risultano spiegate totalmente mediante il formalismo poiché portano informazioni che vanno al di là del formalismo stesso.

Questo linguaggio è stato approntato per risolvere determinati problemi di gestione automatica delle biblioteche.

Una visione critica di come questi problemi vengano risolti mediante il linguaggio LINGEB viene data nei capitoli che seguono.

3.2 Definizione del linguaggio.

Si dà la definizione del linguaggio LINGEB utilizzando il formalismo di Backus-Naur.

```

<comandi> ::= <comando> | <comando> <spazio> <comandi>
<comando> ::= <com> | <COM> <spazio> <frase costruzione> |
               <com> <spazio> <frase aggiornamento> | <com> <spazio> <frase utente>
<com> ::= '/' <fase> '/' <sigla> '.'
<fase> ::= 1 | 2 | 3 | S
<sigla> ::= CTC | CMC | CAL | ATC | ACA | AAL | RUT | SLL | ALT
<frase costruzione> ::= <num> <spazio> <lista parole chiave> | <frase classificazione> |
                       <frase archivio>
<frase aggiornamento> ::= <aggiornamento traduttore> |
                           <aggiornamento classificazione> | <archivio>
<frase utente> ::= <lingua> <spazio> <lista parole chiave> '.' | <num>
<lingua> ::= E | I | F | D
<spazio> ::= ' ' | '\t' <spazio>
<num> ::= <cifra> | <cifra> <num>
<cifra> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
<lista parole chiave> ::= <parola> <spazio> | <parola> |
                          <parola> <spazio> <lista parole chiave>
<parola> ::= <lettera> | <lettera> <parola>

```


<editore>::=<lett>|<lett><editore>

<titolo>::=<vocabolo>|<vocabolo><titolo>

<vocabolo>::=<lett>|<lett><vocabolo>

<lett>::=A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z|
 |1|2|3|4|5|6|7|8|9|,|.|?|!|'|(|)|_

<aggiornamento traduttore>::='1'<cod3><parola><spazio><aggiornamento traduttore>|
 '1' '4'<parola><spazio><classificazione>|
 '0'<cod3><parola><spazio><aggiornamento traduttore>|
 '0' '4'<parola><spazio>

<cod3>::= 1|2|3

<aggiornamento classificazione>::=<parola><spazio><casella> '.'

<casella>::=<cif><num>|<cif><num><spazio><casella>

3.3 Semantica.

<comandi>

Permette di eseguire una fase, o una serie di fasi, per la gestione della biblioteca.

<com>

È uno dei comandi che attiva una determinata fase.

Se <com> è del tipo /1/ attiva una delle tre sottofasi di costruzione CTC, CMC, CAL che indicano rispettivamente la costruzione del traduttore-ordinatore delle chiavi, la costruzione della matrice di classificazione delle chiavi e la costruzione dell'archivio di libri.

Se <com> è del tipo /2/ dà luogo ad una delle tre sottofasi di aggiornamento individuate dalle sigle ATC, ACA, AAL che hanno, rispettivamente, il significato di: aggiornamento del traduttore e delle matrici di classificazione delle chiavi, aggiornamento di un elemento della matrice di classificazione delle chiavi e aggiornamento dell'archivio dei libri.

Se <com> è del tipo /3/ si è nella fase specifica dell'utente che

utilizza la biblioteca e a seconda della sigla, RUT o SLL, viene effettuata la ricerca per parole chiave o la stampa dei libri di una delle classi, che sono state individuate dall'utente mediante la sua ricerca.

/S/ALT. è il comando che si deve usare per interrompere ogni tipo di utilizzazione della biblioteca.

<num>

È un metasimbolo che compare molto di frequente nel linguaggio e a seconda del contesto in cui compare assume significati diversi, pur producendo sempre una stringa di cifre, cioè un numero intero.

Proprio perché assume significati totalmente diversi si è pensato di descriverlo subito dopo le frasi <comando> e <com>, in modo che non generi delle perplessità in chi scorre le produzioni del linguaggio.

Quando compare prima della <lista parole chiave> indica il numero di parole chiave che vengono introdotte per costituire il « thesaurus ».

Come produzione della <frase utente> è il numero della classe della quale l'utente vuole la lista completa dei libri.

Quando compare nella stringa dopo il metasimbolo <classificazione> deve avere lo stesso valore della posizione che occupa nella lista ordinata delle parole chiave in inglese la parola che corrisponde a quella <classificazione>; serve, quindi, come controllo del giusto inserimento delle classificazioni.

Se compare immediatamente prima del metasimbolo <nome classe> è il numero che è stato determinato in fase di classificazione per quell'argomento che è specificato mediante <nome classe>. Invece se precede il metasimbolo <libri> sta ad indicare il numero di classi che si stanno o per creare o per aggiornare, ovviamente questo numero deve essere $\neq 0$. Le informazioni ulteriori di queste classi si ottengono mediante le produzioni delle stringhe a partire dal metasimbolo <libri>.

Infatti le produzioni del metasimbolo <libri> contengono due <num> dove il primo <num> sta ad indicare il numero progressivo che la classe in questione ha in memoria, mentre il secondo <num> il numero di stringhe contenenti informazioni che verranno inserite; questo è sempre un numero pari, poichè per ogni libro vengono inserite due stringhe di informazioni.

Nella stringa che contiene il metasimbolo <col>, il primo <num> indica un valore della classificazione della biblioteca, il secondo <num>

il numero progressivo che è dato al libro quando viene ricevuto dalla biblioteca.

Quando segue <editore> indica l'anno nel quale il corrispondente libro è stato pubblicato.

Infine, quando compare nelle produzioni di <casella> è il numero della colonna della matrice di classificazione, cioè il numero della classe, nella quale si introdurrà <cif>. <lista parole chiave>.

È la frase che indica come vengono introdotte in memoria le parole chiave.

Le parole chiave sono costituite da stringhe di lettere dell'alfabeto inglese, cioè si dà ad esse la struttura di uniterm anche se sono formate da più di una parola. La stringa può essere costituita al massimo di 30 lettere.

<frase classificazione>

Forma la matrice di classificazione.

La matrice ha 73 colonne, cioè ogni colonna corrisponde ad una classe, e tante righe quante sono le parole chiave precedentemente introdotte. Ogni elemento della matrice indica che la parola chiave corrispondente alla riga è propria della classe corrispondente alla colonna se compare un 1, nel caso contrario compare uno 0.

<frase archivio>

Permette di introdurre in memoria le denominazioni delle classi costituenti la biblioteca.

Una volta compiuto questo passo permette o meno di introdurre le informazioni riguardanti i libri di una o più classi.

Infatti le informazioni riguardanti i libri vengono introdotte mediante il metasimbolo <archivio> che non compare in tutte e due le produzioni della frase archivio.

<archivio>

È una frase tipica della costruzione se segue il metasimbolo <clasi>, invece propria del momento dell'aggiornamento se costituisce una delle possibili produzioni della <frase aggiornamento>. In ambedue i momenti permette di costruire le stringhe contenenti le informazioni dell'archivio dei libri.

< titoli >

È il metasimbolo a partire dal quale si formano le stringhe contenenti le informazioni relative ai libri della biblioteca. Per ogni libro si producono due stringhe al massimo di 80 caratteri ciascuna, la prima stringa oltre alla collocazione propria della biblioteca, al nome o ai nomi degli autori porta la casa editrice e l'anno di pubblicazione del libro, la seconda stringa il titolo del libro.

< spazio >

È il metasimbolo che indica le spaziature all'interno delle varie stringhe. Con il simbolo « *̄* » si suole indicare una spaziatura.

4. - IMPLEMENTAZIONE E NORME PER L'USO DEL LINGUAGGIO LINGEB.

4.1 *Introduzione.*

È noto che le biblioteche e i centri di documentazione devono essere organizzati in modo tale da soddisfare all'esigenza di fornire risposte pertinenti alle richieste di documenti in esse contenuti.

Questa ricerca può essere più rapida e precisa se si ricorre all'impiego di un calcolatore, per mezzo del quale si può attuare in modo automatico.

Una tale organizzazione computerizzata deve avere, da un lato, la caratteristica di essere facilmente accessibile all'utente e, dall'altro, essere tale da prestarsi a tutte le possibili fasi di aggiornamento.

Ciò può essere attuato in modo soddisfacente con un sistema di terminali e con l'impiego di un linguaggio di tipo conversazionale.

4.2 *Implementazione del linguaggio LINGEB.*

Il linguaggio LINGEB, definito nel precedente capitolo, unisce alla caratteristica di essere di tipo conversazionale quella di permettere di compiere gli aggiornamenti che si richiedono nella gestione delle biblioteche.

Di esso si è scritto un interprete in linguaggio FORTRAN di cui ora se ne descrive la logica generale di costruzione. È stato necessario

fare delle scelte precise per quanto riguarda l'importanza delle varie fasi da eseguire.

È sembrato, infatti, razionale puntare soprattutto sulla ottimizzazione della fase di ricerca che si prevedeva essere quella maggiormente usata, anche se questo può provocare l'impiego di algoritmi meno ottimizzati nella fase di costruzione.

Vengono ora descritti i vari passi di cui si compone tale interpre-tatore.

Alla fine del presente paragrafo si fornisce un flow-chart che ha il compito di schematizzare tutto il processo.

4.2.1 Costruzione del traduttore simultaneo delle parole chiave.

La ricerca da parte dell'utente avviene per mezzo di un certo numero di parole chiave da esso fornite.

Queste parole chiave possono essere date indifferentemente in una delle quattro seguenti lingue: inglese, italiano, francese, tedesco. Tuttavia tutto l'algoritmo è imperniato sull'uso della lingua inglese.

Bisogna quindi tradurre simultaneamente le richieste formulate in lingue diverse dall'inglese in quest'ultima lingua. Ciò è realizzato disponendo le parole chiave di ogni lingua in ordine alfabetico e costruendo degli accessi alla prima parola di ogni gruppo di parole inizianti per la stessa lettera. Questo per poter giungere più velocemente alle parole chiave.

Quindi se la ricerca avviene in inglese si agisce direttamente sulla lista di parole inglesi, se invece è fatta in una delle altre lingue si passa dalla relativa lista a quella dell'inglese.

4.2.2 Costruzione della matrice di classificazione.

La seconda fase di costruzione consiste nell'associare ad ognuna delle parole chiave della lista ordinata in inglese un vettore i cui elementi sono costituiti da 0 o 1 a seconda che la parola chiave in esame appartenga oppure no all'insieme delle parole chiave che caratterizzano la classe relativa all'ordine dell'elemento del vettore. Questi vettori avranno quindi un numero di elementi pari a quello delle classi in cui è suddivisa la biblioteca e formeranno nel loro complesso la matrice di classificazione.

Questa fase di costruzione può essere attuata solo se si dispone dell'output della fase precedente.

4.2.3 Costruzione dell'archivio dei libri.

In questo archivio vengono memorizzate le informazioni relative ai libri, utili per la loro individuazione.

La sua costruzione viene effettuata memorizzando dapprima le denominazioni delle classi ed inserendo poi, relativamente ad ogni classe, i rispettivi libri.

Come elementi di caratterizzazione dei libri si considerano:

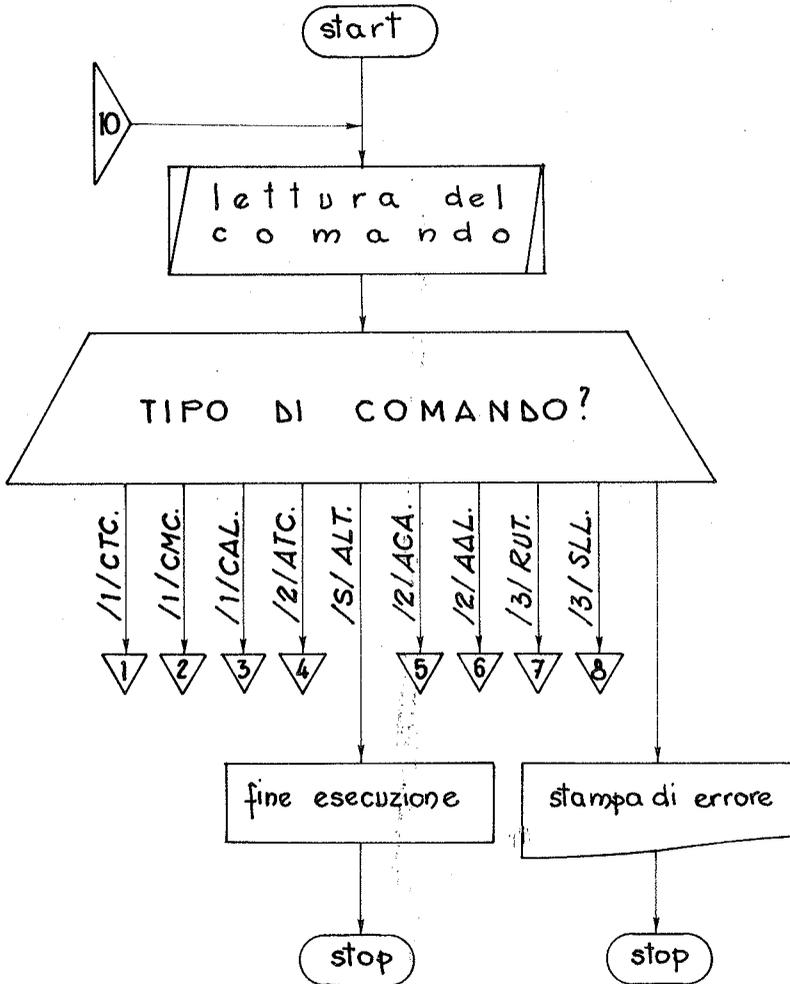
- 1) collocazione nella biblioteca;
- 2) autore/i;
- 3) casa editrice;
- 4) anno di pubblicazione;
- 5) titolo del libro.

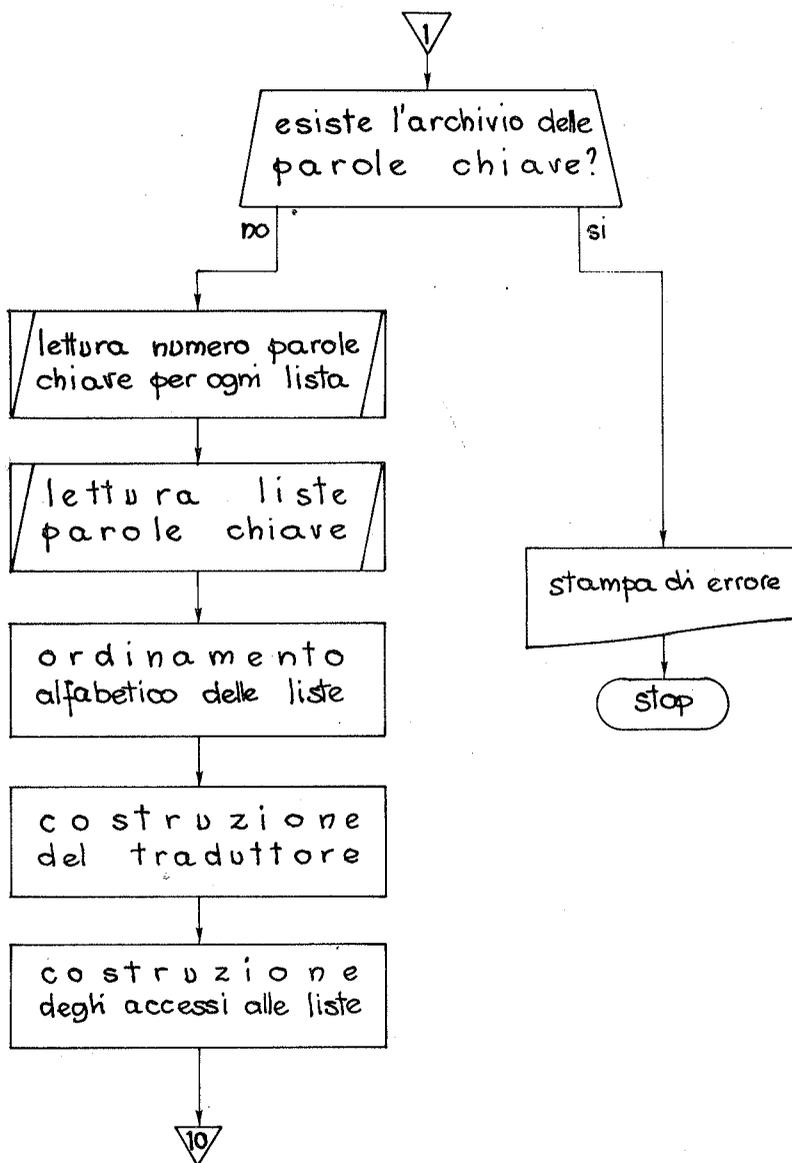
4.2.4 Aggiornamenti.

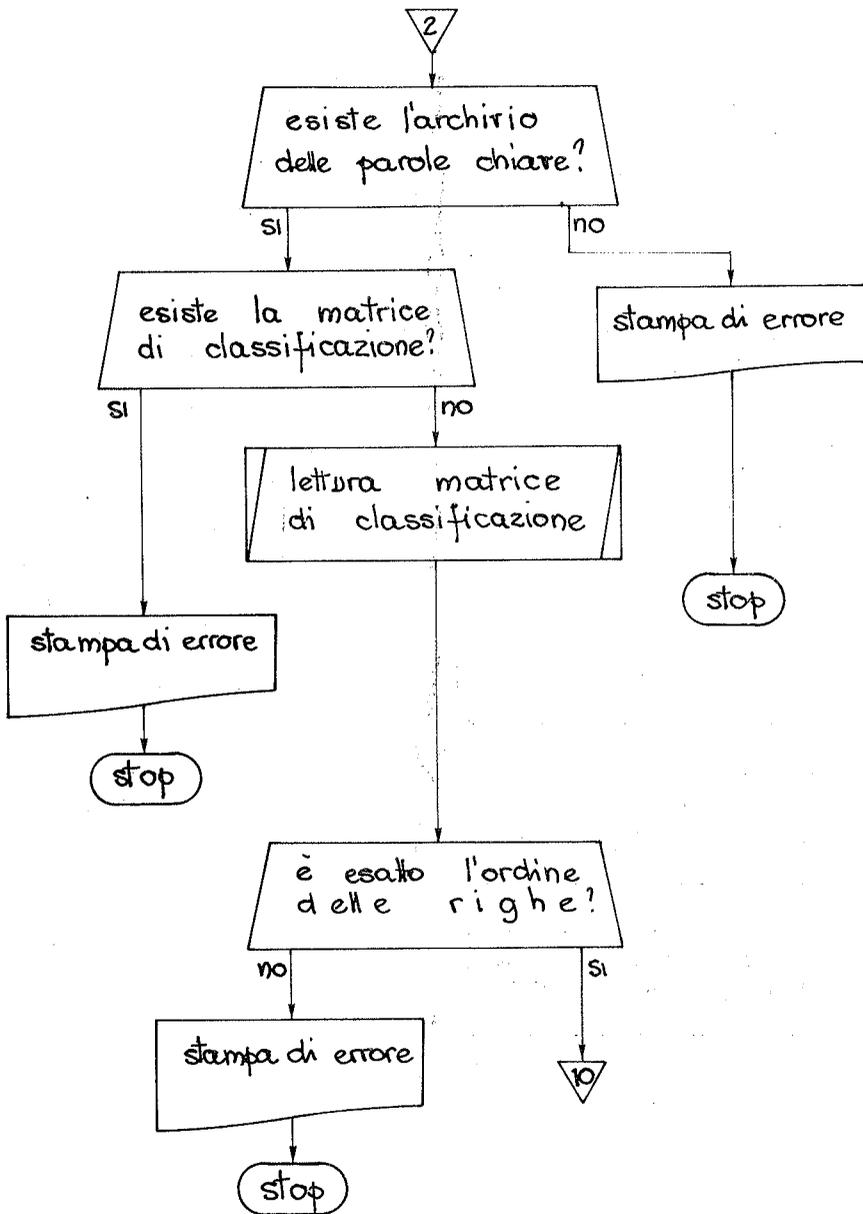
Si possono eseguire i seguenti tipi di aggiornamento:

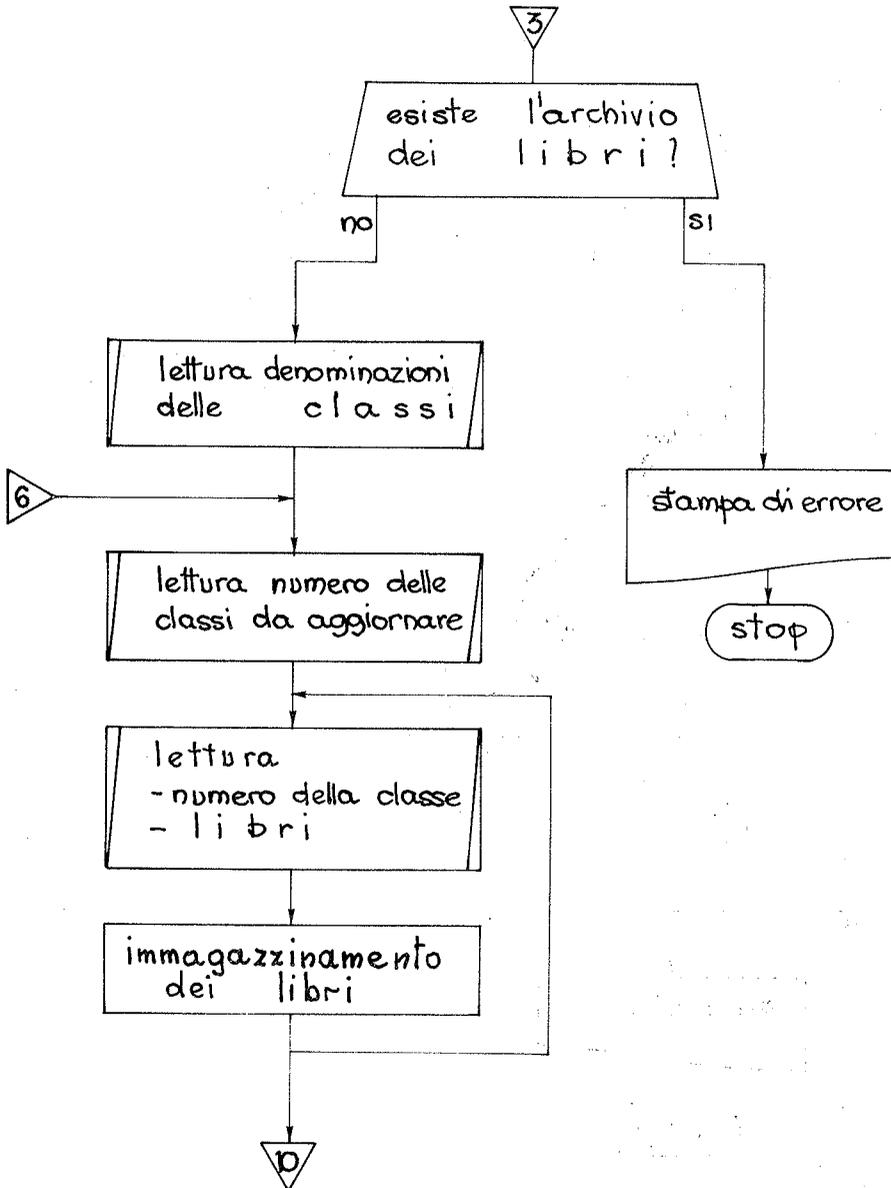
- 1) Inserimento di una nuova parola chiave e del relativo vettore di classificazione. Esso può avvenire solo se la parola chiave da inserire non è già esistente nella lista delle parole chiave in inglese.
- 2) Modifica del contenuto di una parola chiave e del relativo vettore di classificazione.
- 3) Modifica del contenuto di elementi della matrice di classificazione.
- 4) Inserimento di nuovi libri nel relativo archivio.

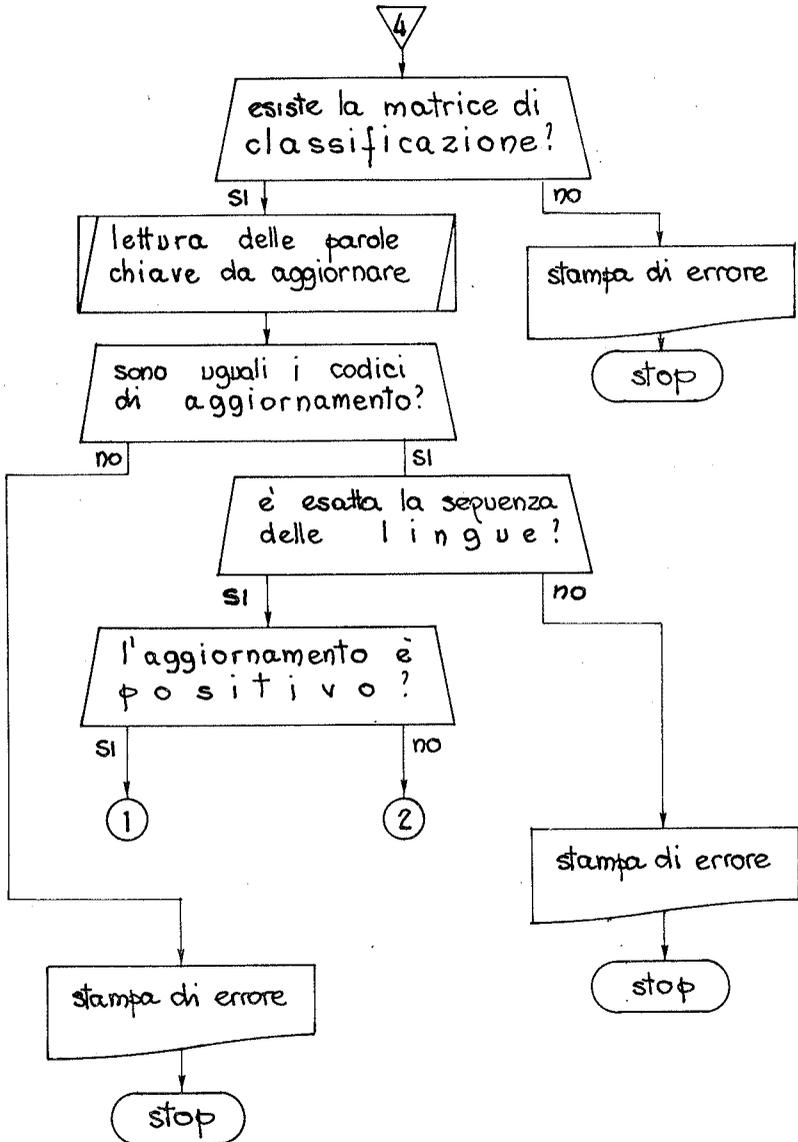
FLOW - CHART

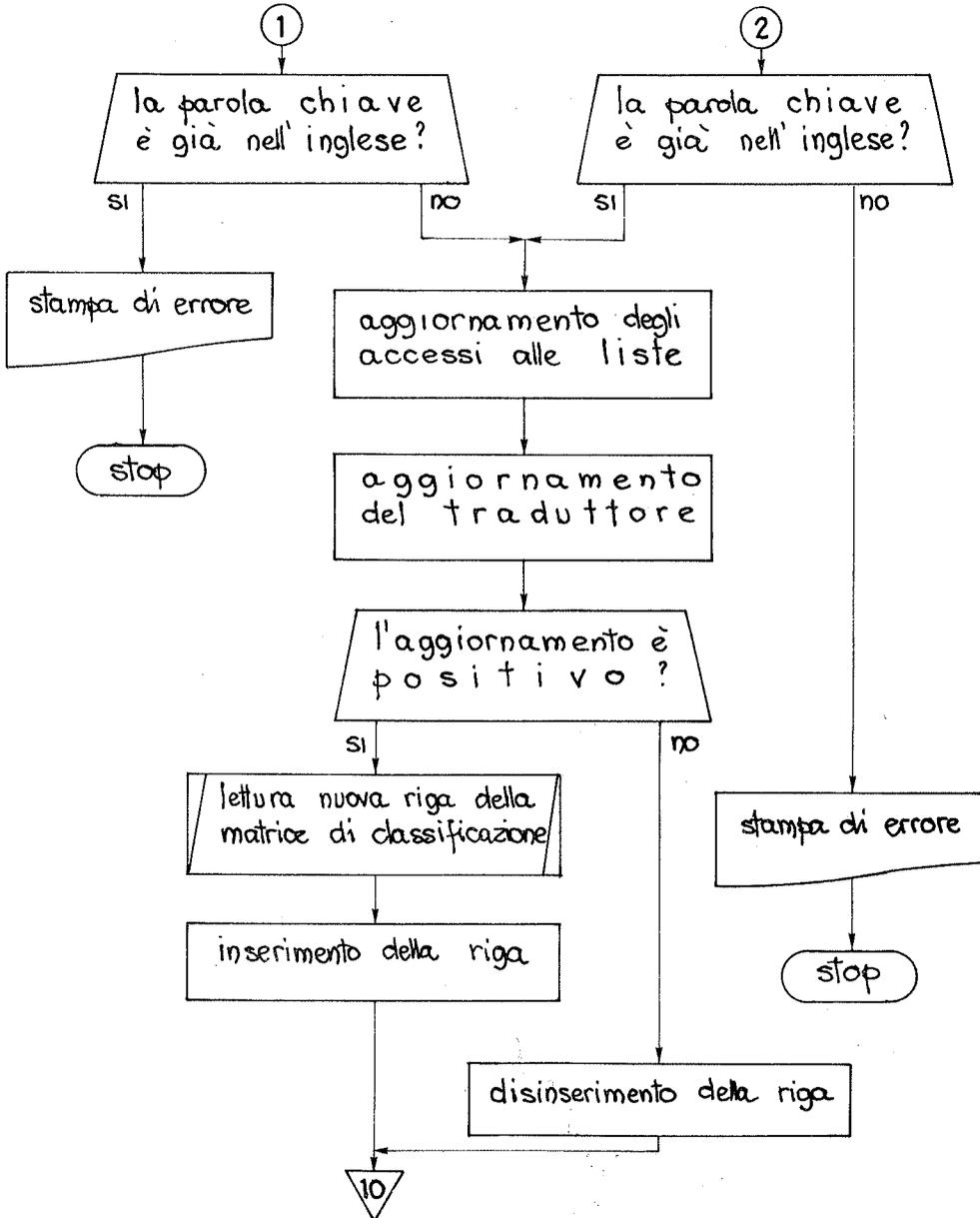


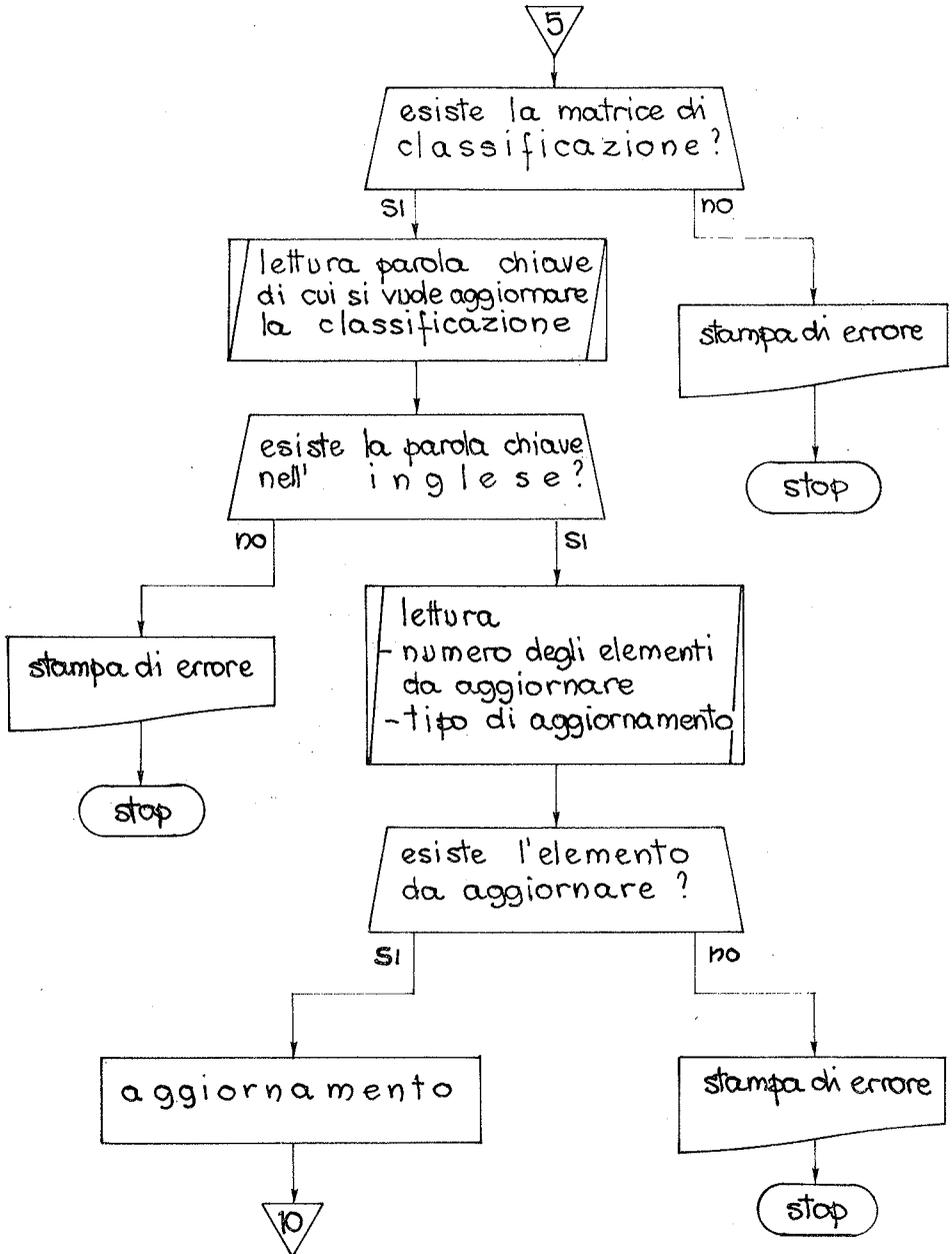


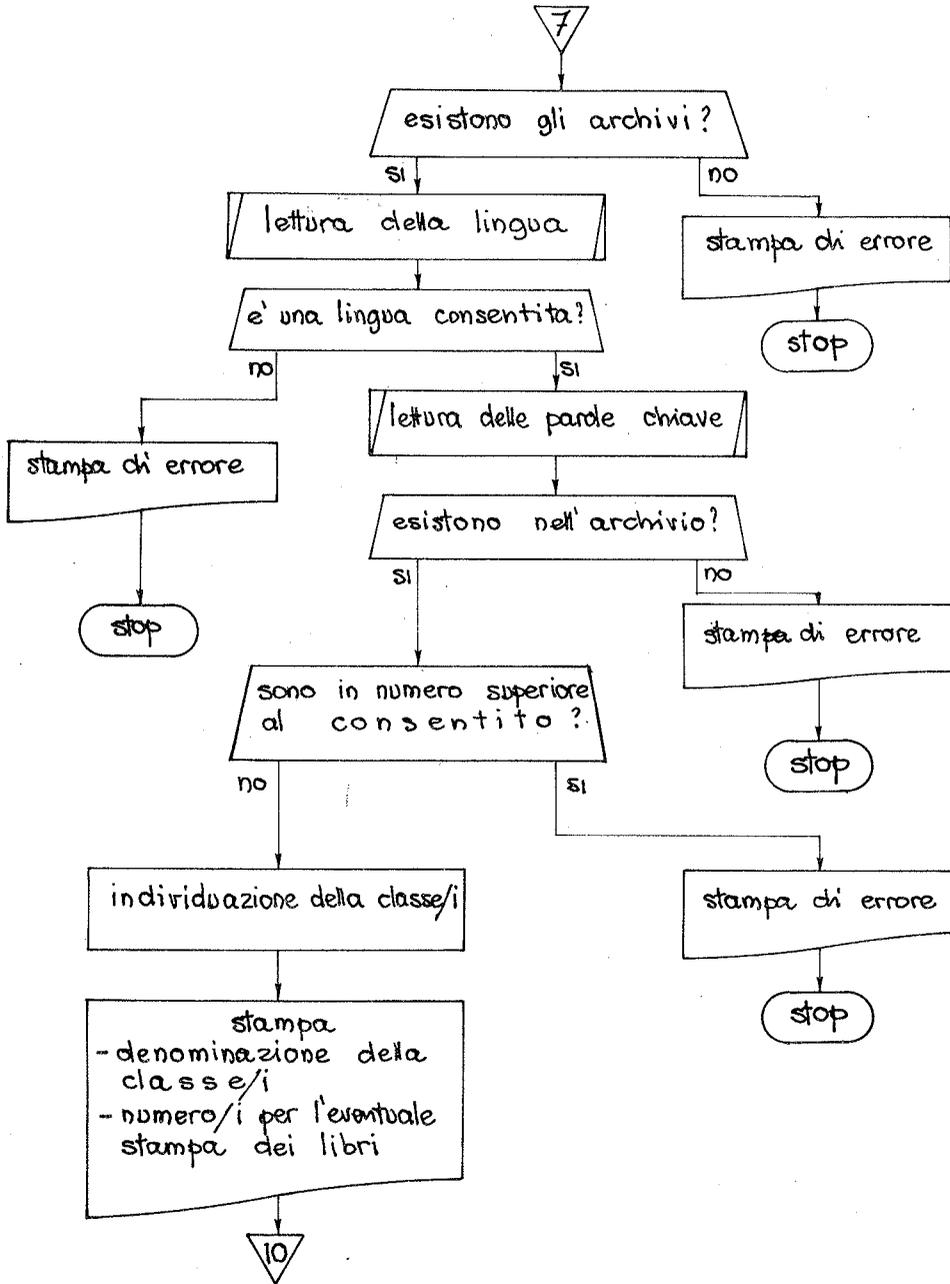


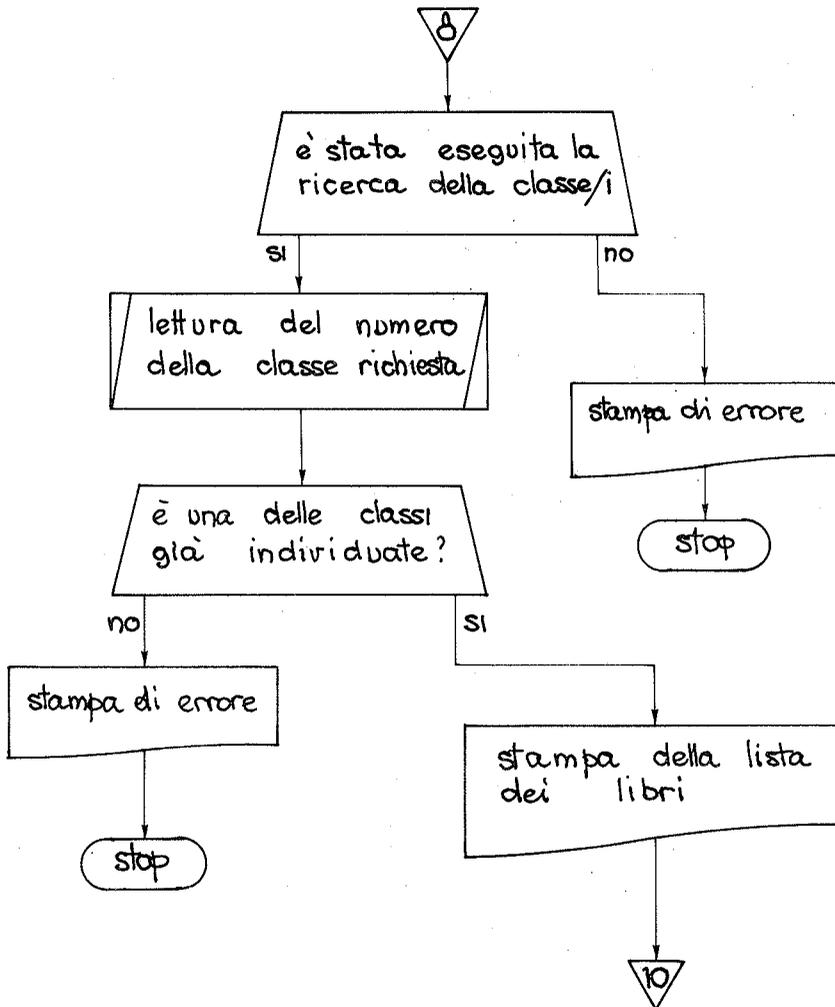












4.3 *Descrizione dei dati di input richiesti.*

Si elencano ora gli input necessari per poter realizzare le varie fasi del linguaggio LINGEB.

Essi sono riportati con riferimento ad un sistema « batch » che è quello che si può usare se non si dispone di un terminale atto alla loro esecuzione in modo conversazionale.

Tuttavia la logica seguita è tale da permettere di passare dall'uno all'altro tipo di gestione in modo assai agevole, purché si possieda detto strumento.

4.3.1 Input per la costruzione del traduttore simultaneo delle parole chiave.

- 1) Comando: /1/CTC.
- 2) Numero delle parole chiave da inserire in ogni singola lingua.
- 3) Gruppi delle parole chiave, rispettivamente nelle lingue: inglese, italiano, francese, tedesco.

4.3.2 Input per la costruzione della matrice di classificazione.

- 1) Comando: /1/CMC.
- 2) Gruppo dei vettori riga della matrice di classificazione con associato il numero d'ordine relativo alla parola chiave in inglese a cui si riferiscono.

4.3.3 Input per la costruzione dell'archivio dei libri.

- 1) Comando: /1/CAL.
- 2) Gruppo delle denominazioni delle classi.
- 3) Gruppi, in numero pari alle classi che si vogliono costruire, così composti:
 - a) numero della classe che si considera e numero di schede con le informazioni relative ai libri,
 - b) schede contenenti le informazioni relative ai libri.

4.3.4 Input per l'inserimento e il disinserimento di una parola chiave.

- 1) Comando: /2/ATC.
- 2) Gruppo di 4 stringhe, ognuna con le seguenti informazioni:

- a) codice del tipo di aggiornamento:
 - 1 per l'inserimento,
 - 0 per l'eliminazione;
 - b) codice della lingua:
 - 1 per la lingua inglese,
 - 2 per la lingua italiana,
 - 3 per la lingua francese,
 - 4 per la lingua inglese.
- 3) Vettore riga della matrice di classificazione, nel solo caso in cui si effettui un inserimento.

4.3.5 Input per l'aggiornamento di un elemento della matrice di classificazione.

- 1) Comando: /2/ACA.
- 2) Parola chiave nella lingua inglese di cui si vogliono modificare uno o più elementi dell'associato vettore riga della matrice di classificazione.
- 3) Gruppo di stringhe contenenti ognuna il valore da inserire ed il numero della classe che individua la posizione di inserimento nel vettore riga. L'ultima stringa deve terminare con il punto.

4.3.6. Input per l'inserimento di nuovi libri nel relativo archivio.

- 1) Comando: /2/AAL.
- 2) Numero delle classi che si vogliono aggiornare.
- 3) Gruppi di stringhe, di numero pari alle classi che si vogliono aggiornare, così composti:
 - a) numero della classe che si considera e numero delle schede da inserire relative ai libri,
 - b) schede contenenti le informazioni relative ai libri.

4.3.7 Input per la ricerca dell'utente.

- 1) Comando: /3/RUT.
- 2) Sigla della lingua in cui si effettua la ricerca:
 - E per la lingua inglese,
 - I per la lingua italiana,
 - F per la lingua francese,
 - D per la lingua tedesca.

3) Gruppo delle parole chiave che si usano per la ricerca, tutte nella lingua prescelta. L'ultima parola chiave deve essere seguita dal punto.

4.3.8 Input per la stampa delle informazioni relative ai libri della classe/i individuata.

- 1) Comando: /3/SLL.
- 2) Numero della classe, già in precedenza individuata, di cui si richiede la stampa delle informazioni relative ai libri.

4.3.9 Input per la fine dell'esecuzione.

Per giungere in qualsiasi momento alla fine dell'esecuzione basta usare il comando: /S/ALT.

4.4 *Esempio di applicazione.*

L'applicazione di tale metodo di gestione ad un caso concreto è attuata con riferimento alla biblioteca della facoltà di Scienze Statistiche dell'Università di Padova.

Detta biblioteca è suddivisa in tre blocchi:

- 1) Riviste,
- 2) Monografie (libri, estratti, opuscoli),
- 3) Documentazione ufficiale.

La realizzazione è rivolta al blocco delle monografie, che sono suddivise in gruppi e all'interno dei gruppi in classi e sottoclassi.

Essa, tuttavia, è limitata alle classi che sono in numero di 73.

Inoltre, per ragioni di semplicità, si considerano solo classi relative ai due seguenti gruppi:

- a) elaborazione automatica dei dati,
- b) statistica metodologica.

Alla fine del seguente capitolo si forniscono le liste delle parole in ordine alfabetico nelle lingue: inglese, italiano, francese, tedesco; l'elenco dei libri utilizzati nell'esemplificazione, il listing del programma principale e delle subroutine usate, e due esempi di ricerca effettuati dall'utente.

ELENCO DELLE PAROLE CHIAVE

L I N G U A I N G L E S E

A

ABSOLUTE PROGRAM
 ABSTRACTION
 ADVANTAGES OF FACTORIAL EXPERIM.
 AGGREGATE
 ALLOCATABLE DEVICE
 ANALYSIS OF COVARIANCE
 ANALYSIS OF VARIANCE
 AREA GRAPH
 ARRAY
 ASSEMBLER
 ATTRIBUTE

B

BACKUS NAUR FORM
 BALANCED INCOMPLETE BLOCKS
 BARTLETT TEST
 BAYES ESTIMATION
 BAYES THEOREM
 BETA DISTRIBUTION
 BIAS
 BIENAYME TCHEBICHEFF THEOREM
 BINOMIAL R.V.
 BOTTOM UP ANALYSIS

C

CARTOGRAM
 CENTRAL LIMIT THEOREM
 CHARACTERISTIC FUNCTION
 CHOICE OF FACTOR
 CHOMSKY CLASSIFICATION
 CLUSTER SAMPLING
 COCHRAN TEST
 CODE GENERATION
 COEFFICIENT OF VARIATIONS
 COEFFICIENT OF CORRELATION
 COMPILER
 COMPLETELY RANDOMIZED DESIGN
 COMPONENT OF VARIATION

CONNECTION
 CONTINGENCY QUADRATIC INDEX
 CONTINGENCY TABLE
 CONTROL CARD
 CONTROL TREATMENT
 CONVERGENCE IN DISTRIBUTION
 CONVERGENCE IN PROBABILITY
 CONVERGENCE OF MOMENTS
 CUMULATIVE DISTRIBUTION FUNCTION
 CURVE OF CONCENTRATION

D

DEFINITION OF MAIN EFFECT
 DESIGN OF LEXIS
 DESCRIMINATOR ANALYSIS
 DISK PAK
 DOUBLE SAMPLING
 DRUM

E

END OF FILE
 ESTIMATES OF VARIANCES
 ESTIMATES OF MEANS
 ESTIMATION PROPRIETIES
 EXPONENTIAL DISTRIBUTION

F

FILE
 FISHER SNEDECOR TEST
 FLOYD PRODUCTIONS
 FRACTIONAL REPLICATION

G

GAMMA DISTRIBUTION
 GENERATING FUNCTION

H

HASH TABLES
 HISTOGRAMM
 HOTELLING TEST
 HYPERGEOMETRIC R.V.
 HYPOTHESIS

I

INDIPENDENCE CHJ SQUARED TEST
 INQUIRY
 INTERACTION
 INTERVAL ESTIMATION

J

JOB CONTROL
 JOB

K

KOLMOGOROV SMJRN OV TEST

L

LATIN SQUARES
 LAW OF LARGE NUMBERS
 LAW OF RANDOMISATION
 LEAST SQUARES METHOD
 LEXICAL ANALYSIS
 LIKELIHOOD RATIO
 LINEAR HYPOTHESIS
 LINKAGE EDITOR
 LISTS
 LOADER
 LOGARITHMIC NORMAL DISTRIBUTION

M

MAGNETIC TAPE
 MANN WHITNEY TEST
 MAXIMUM LIKELIHOOD METHOD
 MEAN
 MEDIAN
 METHOD OF MOMENTS
 MODALITY

MODE

MOMENTS R.V.
 MONITOR
 MORTARA INDEX
 MOST SELECTIVE CONFIDENCE INT.
 MULTI,COLLINEARITY

N

NEYMAN PEARSON LEMMA
 NOMINAL SCALE
 NONTERMINAL SYMBOL
 NORMAL DISTRIBUTION

O

OBSERVATIONS
 OPERATING SYSTEM
 OPERATOR PRECEDENCE GRAMMARS
 ORDINAL SCALE
 OVERLAY PROGRAM

P

PERCENTILS
 PICTOGRAM
 PLEXES
 POINT ESTIMATION
 POISSON R.V.
 POLISH
 POPULATION
 POSTERIO PROBABILITY
 PRIMARY SAMPLING UNITS
 PRIOR PROBABILITY
 PROBABILITY PROPORTIONAL TO SIZE
 PROBABILITY DENSITY FUNCTION
 PROBABILITY FUNCTION
 PRODUCTIONS
 PROGRAMMING LANGUAGE
 PROPERTY OF A TEST

Q

QUEUES AND STACKS

R

RANDOM ACCESS
 RANDOM FILE
 RANDOMIZATION
 RAPPORT SCALE
 RATIO ESTIMATE
 RATIO SCALE
 RECORD
 REGRESSION CURVE
 RESPONSE SURFACE
 RESULTS ARRANGEMENT
 ROTATION EXPERIMENTS
 R.V. DEFINITION
 R.V. TRANSFORMATION

S

SAMPLING FRACTION
 SAMPLING MEAN
 SAMPLING UNIT
 SAMPLING VARIANCE
 SAMPLING WITH REPLACEMENT
 SAMPLING WITHOUT REPLACEMENT
 SCRUTINY
 SEMANTIC ANALYSIS
 SEQUENTIAL FILE
 SEQUENTIAL TEST OF WALD
 SEQUENTIAL ACCESS
 SEQUENTIAL ANALYSIS
 SOURCE PROGRAM
 STATISTICAL UNIT
 STATISTIC INTERPOLATION
 STOCHASTIC PROCESSES
 STRATIFICATION
 STRINGS
 STUDENT DISTRIBUTION
 STUDENT TEST
 SUBSAMPLING

SUPERVISOR
 SYNTAX ANALYSIS
 SYSTEMATIC SAMPLING

T

TABLES
 TABLE
 TAPE LABEL
 TERMINAL SYMBOL
 THREE STAGE SAMPLING
 TOP DOWN ANALYSIS
 TREES
 TYPES OF ERRORS INSURVEYES

U

UNIFORM R.V.
 UNITS EXPERIMENTAL
 UTILITY

V

VARIABLE
 VARIANCE
 VECTORS

W

WILCOXON TEST

ELENCO DELLE PAROLE CHIAVE

L I N G U A I T A L I A N A

<p>A</p> <p>ACCESSO DIRETTO ACCESSO SEQUENZIALE AGGREGATO ALBERI ALLOCATABLE DEVICE ANALISI ASCENDENTE ANALISI DELLA COVARIANZA ANALISI DELLA VARIANZA ANALISI DISCENDENTE ANALISI DISCRIMINATORIA ANALISI LESSICALE ANALISI SEMANTICA ANALISI SEQUENZIALE ANALISI SINTATTICA ASSEMBLATORE ASTRAZIONE</p> <p>B</p> <p>BIAS BLOCCHI INCOMPLETI</p> <p>C</p> <p>CAMPIONAMENTO A DUE STADI CAMPIONAMENTO A TRE STADI CAMPIONAMENTO SISTEMATICO CAMPIONAMENTO CON RIPETIZIONE CAMPIONAMENTO A GRAPPOLO CAMPIONAMENTO SENZA RIPETIZIONE CARATTERE QUALITATIVO CARATTERE QUANTITATIVO CARTOGRAMMA CASUALIZZAZIONE CLASSIFICAZIONE DI CHOMSKY CODE E PILE COEFFICIENTE DI CORRELAZIONE COEFFICIENTE DI VARIAZIONE COMPILATORE</p>	<p>COMPONENTE DI VARIAZIONE CONNESSIONE CONVERGENZA DEI MOMENTI CONVERGENZA IN DISTRIBUZIONE CONVERGENZA IN PROBABILITA' CURVA DI CONCENTRAZIONE CURVA DI REGRESSIONE</p> <p>D</p> <p>DEFINIZIONE DI V.C. DEFINIZIONE DELL'EFFETTO PRINCIP. DISCO MAGNETICO DISTRIBUZIONE NORMALE DISTRIBUZIONE ESPONENZIALE DISTRIBUZIONE GAMMA DISTRIBUZIONE LOGNORMALE DISTRIBUZIONE BETA DISTRIBUZIONE T DI STUDENT</p> <p>E</p> <p>ESPERIMENTI DI ROTAZIONE</p> <p>F</p> <p>FILE DIRETTO FILE SEQUENZIALE FILE FINE DEL FILE FORMA DI BACKUS NAUR FRAZIONE DI CAMPIONAMENTO FUNZIONE CARATTERISTICA FUNZIONE DI RIPARTIZIONE FUNZIONE DI DENSITA' DI PROBABILITA' FUNZIONE DI PROBABILITA' FUNZIONE GENERATRICE</p> <p>G</p>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

GENERAZIONE DEI CODICI	N
GRAFICO AD AREE	
GRAMMATICA DI OPERATORI A PRECED. NASTRO ETICHETTATO	
I	O
IDEOGRAMMA	ORDINAMENTO DEI RISULTATI
INDICE DI CONTINGENZA DEL MORTARA	OSSERVAZIONI
INDICE QUADRATICO DI CONTINGENZA	
INTERAZIONE	P
INTERPOLAZIONE STATISTICA	
INTERVALLO DI CONFIDENZA OTTIMALE	PADELLA
IPOTESI LINEARI	PERCENTILI
IPOTESI STATISTICHE	PIANO ASSOLUTAMENTE CASUALE
ISTOGRAMMA	PLESSI
	POLACCA
	POPOLAZIONE
J	PROBABILITA' A POSTERIORI
JOB DI CONTROLLO	PROBABILITA' A PRIORI
JOB	PROBABILITA' PROPORZ. ALLA MIS.
	PROCESSI STOCASTICI
	PRODUZIONI DI FLOYD
	PRODUZIONI
L	PROGRAMMA D'UTILITA'
LEGGE DEI GRANDI NUMERI	PROGRAMMA OVERLAY
LEGGE EMPIRICA DEL CASO	PROGRAMMA RILOCABILE
LEMMA DI NEYMAN E PEARSON	PROGRAMMA SORGENTE
LINGUAGGIO DI PROGRAMMAZIONE	PROPRIETA' DI UN TEST
LINKAGE EDITOR	PROPRIETA' DELLA STIMA
LISTE	
LOADER	C
	QUADRATI LATINI
M	
MATRICE	R
MEDIA CAMPIONARIA	
MEDIANA	
MEDIA	RAPPORTO DI MAX VEROSIMIGLIANZA
METASIMBOLO	REGISTRAZIONE
METODO DEI MINIMI QUADRATI	REPLICAZIONE FRAZIONARIA
METODO DEI MOMENTI	RILEVAZIONE
METODO DELLA MAX. VEROSIMIGLIANZA	
MODALITA'	S
MODA	
MOMENTI DI V.C.	SCALA INTERVALLI
MONITOR	SCALA NOMINALE
MULTICOLLINEARITA'	

SCALA ORDINALE
 SCALA RAPPORTO
 SCELTA DEL FATTORE
 SCHEDE CONTROLLO
 SCHEMA DI LEXIS
 SIMBOLO TERMINALE
 SISTEMA OPERATIVO
 SOTTOCAMPIONE
 SPOGLIO
 STIMA DELLA VARIANZA
 STIMA DELLA MEDIA
 STIMA DI BAYES
 STIMA INTERVALLARE
 STIMA PER RAPPORTO
 STIMA PUNTUALE
 STRATIFICAZIONE
 STRINGA
 SUPERFICIE DI RISPOSTA
 SUPERVISORE

T

TABELLA DI CONTINGENZA
 TABELLA
 TABELLA HASH
 TABELLE
 TAMBURNO
 TEOREMA DEL LIMITE CENTRALE
 TEOREMA DI BAYES
 TEOREMA DI BIENAYME TCHEBYCHEFF
 TEST CHI QUADRATO DI INDIPENDENZA
 TEST DI BARTLETT

TEST DI COCHRAN
 TEST DI KOLMOGOROV E SMIRNOV
 TEST DI MANN WHITNEY
 TEST DI WILCOXON
 TEST F DI FISHER E SNEDECOR
 TESTS SEQUENZIALI WALDIANI
 TEST T DI HOTELLING
 TEST T DI STUDENT
 TIPI DI ERRORE NELL'INDAGINE
 TRASFORMATE DI V.C.
 TRATTAMENTO DI CONTROLLO

V

UNITA' DI CAMPIONAMENTO
 UNITA' PRIMARIA DI CAMP.
 UNITA' SPERIMENTALI
 UNITA' STATISTICA

V

VANTAGGI DEGLI ESPERIMENTI FATT.
 VARIANZA CAMPIONARIA
 VARIANZA
 V.C. BINOMIALE
 V.C. DI POISSON
 V.C. IPERGEOMETRICA
 V.C. RETTANGOLARE
 VETTORI

ELENCO DELLE PAROLE CHIAVE

L I N G U A F R A N C E S E

A

ABSTRACTION
 ACCESS DIRECT
 ACCESSE LIVRE DE SEQUENCES
 AGREGAT
 ALLOCATABLE DEVICE
 ANALYSE ASCENDANTE
 ANALYSE DE LA COVARIANCE
 ANALYSE DE LA VARIANCE
 ANALYSE DESCENDANTE
 ANALYSE DISCRIMINANTE
 ANALYSE LEXICALE
 ANALYSE SEMANTIQUE
 ANALYSE SEQUENTIEL
 ANALYSE SYNTAXIQUE
 ARBRES
 ASSEMBLEUR
 AVANTAGES DES EXPERIENCES

B

BIAS
 BLOCS INCOMPLETES

C

CARACTERE QUALITATIF
 CARACTERE QUANTITATIF
 CARRÉS LATIN
 CARTE STATISTIQUE
 CHOIX DU FACTEUR
 CLASSIFICATION DE CHOMSKY
 COEFFICIENT DE VARIATION
 COEFFICIENT DE CORRELATION
 COMPILATEUR
 COMPOSANTE DE VARIATION
 CONNEXION
 CONVERGENCE EN DISTRIBUTION
 CONVERGENCE EN PROBABILITE
 CONVERGENCE DES MOMENTS
 COURBE DE CONCENTRATION

COURBE DE REGRESSION

D

DEFINITION DE V.A.
 DEFINITION DU EFFET PRINCIPAL
 DEPOUILLEMENT
 DIAGRAMME FIGURATIF
 DISKPAK
 DISQUE MAGNETIQUE
 DISTRIBUTION GAMMA
 DISTRIBUTION BETA
 DISTRIBUTION EXPONENTIELLE
 DISTRIBUTION NORMALE
 DISTRIBUTION DE STUDENT
 DISTRIBUTION LOGNORMALE

E

ECHELLE A INTERVALLES
 ECHELLE DE RAPPORT
 ECHELLE NOMINALE
 ECHELLE ORDINALE
 END OF FILE
 ESTIMATION DE BAYES
 ESTIMATION PAR INTERVALLE
 ESTIMATION DE LA VARIANCE
 ESTIMATION PONCTUELLE
 ESTIMATION DE LA MOYENNE
 ESTIMATION PAR RAPPORT
 EXPERIENCES DE ROULEMENT

F

FICHE DE CONTROL
 FICHER
 FILE DIRECT
 FILE LIVRE DE SEQUENCES
 FONCTION CARACTERISTIQUE
 FONCTION DE REPARTITION
 FONCTION DE DENSITE DE PROBABILITE

FONCTION GENERATRICE
 FORME DE BACKUS NAUR
 FRACTION DE SONDAGE

H

HISTROGRAMME
 HYPOTHESES LINEARS
 HYPOTHESES STATISTIQUES

I

INDICE DE CONTINGENCE DU MORTARA
 INDICE QUADRATIQUE DE CONTINGENCE
 INTERACTION
 INTERPOLATION STATISTIQUE
 INTERVALLE DE CONFIANCE OPTIMALE

J

JOB CONTROL
 JOB

L

LANGAGES DE PROGRAMMATION
 LEMME DE NEYMAN ET PEARSON
 LINKAGE EDITOR
 LISTES
 LOADER
 LOI DES GRANDS NOMBRES
 LOI EMPIRIQUE DU CAS

M

MATRICES
 MEDIANE
 METHODES DES MOINDRES CARRES
 METHODES DES MOMENTS
 METHODE DU MAX VRAISEMBLANCE
 MODALITE
 MODE
 MOMENT DE V.A.
 MONITOR
 MOYENNE D'ECHANTILLONS
 MOYENNE
 MULTICOLLINEARITE

O

OBSERVATIONS
 ORDONNANCE DES RESULTATS

P

PLAN ABSOLUMENT ALEATOIRE
 PLEXES
 POLONAISE
 POPULATION
 POURCENTIL
 PROBABILITE A PRIORI
 PROBABILITE PROPORTIONELLE A LA M.
 PROBABILITE A POSTERIORI
 PROCESSUS STOCHASTIQUES
 PRODUCTIONS
 PRODUCTIONS DE FLOYD
 PROGRAMME ABSOLUE
 PROGRAMME D'UTILITE
 PROGRAMME OVERLAY
 PROGRAMME SOURCE
 PROPRIETE DE L'ESTIMATION
 PROPRIETES DU TEST

Q

QUEUES ET PILES

R

RANDOMISATION
 RAPPORT DE MAX DE VRAISEMBLANCE
 RECORD
 REPLIQUE FRACTIONNAIRE
 RILEVATION

S

SCHEMA DE LEXIS
 SONDAGE A DEUX DEGRES
 SONDAGE A TROIS DEGRES
 SONDAGE AVEC REPETABILITE
 SONDAGE EN GRAPPES
 SONDAGE SANS REPETABILITE
 SONDAGE SYSTEMATIQUE
 SOUS ECHANTILLON
 STRATIFICATION
 STRINGS
 SUPERVISEUR

SURFACE DE REPONSE
 SYMBOL NON TERMINAL
 SYMBOL TERMINAL
 SYSTEME OPERATIF

THEOREME DE BIENAYME TCHEBYCHEFF
 THEOREME DE BAYES
 TRAITEMENT DE CONTROLE
 TRANSFORMES DE V.A.
 TYPES D'ERREUR DANS L'ENQUETE

T

TABLEAU DE CONTIGENCE
 TABLEAUX HASH
 TABLEAUX
 TABLEAU
 TAMBOUR
 TAPE LABEL
 TEST DE BARTLETT
 TEST DE COCHRAN
 TEST DE KOLMOGOROV ET SMIRNOV
 TEST DE MANN WHITNEY
 TEST DE WILCOXON
 TEST DE FISHER ET SNEDECOR
 TEST KHI CARRE D'INDIPENDENCE
 TEST SEQUENTIELS DE WALD
 TEST DE HOTELLING
 TEST DE STUDENT
 THEOREME CENTRAL LIMITE
 THEOREME DE BIENAYME TCHEBYCHEFF

U

UNITE DE SONDAGE
 UNITES EXPERIMENTALES
 UNITES PRIMAIRES DE SONDAGE
 UNITE STATISTIQUE

V

V.A. DE BERNOULLI
 V.A. DE POISSON
 V.A. HYPERGEOMETRIQUE
 V.A. RECTANGULAIRE
 VARIANCE D'ECHANTILLONS
 VARIANCE
 VECTEURS

ELENCO DELLE PAROLE CHIAVE

L I N G U A T E D E S C A

<p>A</p> <p>ABNAHME ABSOLUT STOCHASTISCHEFLACHE ABSTEIGEND ANALYSE ABSTRAKTION AGGREGAT ALLOCATABLE DEVICE ANTWORTFLACHE ASSEMBLATOR AUFSTEIGEND ANALYSE</p> <p>B</p> <p>BAKUS NAUR FORM BAND MIT ETIKETTE BARTLETT TEST BAUME BAYES THEOREM BEENDBAR SYMBOL BEHANDLUNG DER KONTROLLE BEOBACHTUNGEN BETA VERTEILUNG BEVOLKERUNG BIAS BIENAYME TCHEBYCHEFF THEOREM BILDUNGS ELEMENT DER ANDERUNG BINOMIALE STOCHASTISCHE VARIABLE BRUCH WIEDERHOLUNG</p> <p>C</p> <p>CHARAKTERISTISCHE FUNKTION CHI QUADRAT TEST CHOMSKY KLASSIFIZIRUNG COCHRAN TEST CONTIGENZ TABELLE</p> <p>D</p> <p>DEFINITION VON STOCHASTISCHE V.</p>	<p>DEFINITION DER HAPTEFFEKT DICHTESTE DIREKT FILE DIREKT ZUGANG DISKRIMINANZ ANALYSE DREISTUFIGE STICHPROBE</p> <p>E</p> <p>EIGENHEIT VON DER SCHATZER EINHEIT DER ERSTEN AUSWAHLSTUFE ENDE FILE ERZEUGENDE FUNKTION EXPERIMENTELLE INHEIT EXPONENTIELLE VERTEILUNG</p> <p>F</p> <p>FILE FISHER SNEDECOR TEST FLASCHEN DIAGRAMM FLOYD PRODUKTION FOLGE TEST VON WALD FRACHTFUHRER</p> <p>G</p> <p>GAMMA VERTEILUNG GESETZ DER GROSSEN ZAHLEN</p> <p>H</p> <p>HALFTE SYMBOL HASH TABELLE HISTOGRAMM HOTELLING TEST HYPERGEOMETRISCHE S.V.</p> <p>I</p>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

IDEOGRAMM

J

KARTOGRAMM
 KLUMPEN STICHPROBE ZWEISTUFIGE
 KLUMPEN STICHPROBEVERFAHRE
 KODE GERATION
 KOLMOGOROV SMIRNOV TEST
 KONTINGENZ INDEX VON MORTARA
 KONTROLLE JOB
 KONTROLLKARTE
 KONVERGENZ WAHRSCHEINLICHKEIT
 KONZENTRAZIONS KURVE
 KORRELATIONS KOEFFIZIENT
 KOVARIANZ ANALYSE

L

LATEINISCHES QUADRAT
 LESIKALISCH ANALYSE
 LEXIS SCHEMA
 LINEARE HYPOTHESE
 LINKAGE EDITOR
 LOADER
 LOGNORMALE VERTEILUNG

M

MAGNETISCH SCHEIBE
 MANN WHITNEY TEST
 MATRIJZE
 M. DER MAXIMALE MUTMASSLICHKEIT
 MEDIAN WERT
 METHODE DER KLEINSTEN QUADRATE
 METHODE DER MOMENTE
 MITTEL DER STICHPROBEN VERTEILUNG
 MITTELSCHATZ FUNKTION
 MITTEL
 MODALITAT
 MOMENT KONVERGENZ
 MOMENT VON STOCHASTISCHE V.
 MONITOR
 MULTICOLLINEARITAT

N

NAMENTLICH SKALE

NEYMAN PEARSON STICHWORT
 NORMAL VERTEILUNG
 NUTZEN PROGRAMM

O

OBERLEITER
 OPERATEUR VORTRITT GRAMMATIK
 OPERATIV SYSTEM
 ORDINAL SKALE
 ORDNER DER RESULTAT
 OVERLAY PROGRAM

P

PERZENTILE
 PFANNE
 PLEXUS
 POISSON S.V.
 POLONASE
 PRODUKTION
 PUNKSCHATZUNG

Q

QUADRATISCHE KONTINGENZ INDEX
 QUALITATIVE MERKMAL
 QUANTITATIVE MERKMAL
 QUELLE PROGRAMM

R

RECHTECKIGE S.V.
 REGISTRIERUNG
 REGRESSIONS KURVE
 RELATIVER STICHPROBEN UMFANG
 ROTATION PROBEN

S

SCHATZUNG VON BAYES
 SCHICHTEN BILDUNG
 SCHNURRIEMEN
 SCHWANKUNG KOEFFIZIENT
 SCHWANZ UND STOSS
 SEMANTIK ANALYSE
 SEQUENZ ANALYSE

SEQUENZ FILE
 SEQUENZ ZUGANG
 SPRACHEN PROGRAMM
 STATISTISCHE HYPOTHESE
 STATISTISCHE EINSCHALTUNG
 STATISTISCHE EINHEIT
 STICHPROBEN FEHLER
 STICHPROBEN MIT ZURUCKLEGEN
 STICHPROBEN EINHEIT
 STICHPROBEN OHNE ZURUCKLEGEN
 STOCHASTISCHER PROZESS
 STOCHASTISCHE GESETZ
 STREIFEN
 STUDENT TEST
 SYNTAKTISCH ANALYSE
 SYSTEMATICH STICHPROBE

T

TABELLE
 TAFEL
 TEST EIGENSCHAFT
 THEOREM ZENTRALER GRENZWERTSATZ
 TRENNSCHEIDUNG KONFIDENZ BEREICH
 TROMMEL
 T VERTEILUNG

U

UNFORMUNG VON STOCHASTISCHE V.
 UNTERSTICHPROBE
 UNVOLLENDET BLOKADEN

V

VARIANZ ANALYSE
 VARIANZ DER STICHPROBEN VERTEIL
 VARIANZ SCHATZ FUNKTION
 VERGLEICH SCHATZUNG
 VERHALTNISMÄSSIG WAHRSCHEINLIC.
 VERHALTNIS SKALE
 VERLESUNG
 VERTEILUNG KONVERGENZ
 VERTEILUNG FUNKTION
 VARIANZ
 VORTEILE DER FACTORIELLE PROBEN

W

WAHLDER ELEMENT
 WAHRSCHEINLICHKEIT IM VORAUSS
 WAHRSCHEINLICHKEIT DICHTEN
 WAHRSCHEINLICHKEIT FUNKTION
 WAHRSCHEINLICHKEIT A POSTERIORI
 WECHSEL WIRKUNG
 WIEDERVERMIETEN PROGRAMM
 WILCOXON TEST

Z

ZUFALL
 ZUSAMMENHANG DER MAXIMALEN MUTM.
 ZUSAMMENHANG
 ZUSAMMENTRAGER
 ZWISCHENRAUM SCHATZUNG
 ZWISCHENRAUM SKALE

ELENCO DEI LIBRI UTILIZZATI NELL'ESEMPLICAZIONE

21 SISTEMI OPERATIVI

-21 -9	BOOTH A.D.	BUTTERWORTHS	60
	PROGRES IN AUTOMATION		
R -21 -4	C.N.U.C.E. 11	U.PISA	
	CONCETTI E POSSIBILITA' DEL SISTEMA OPERATIVO OS/360		
R -21 -3	C.N.U.C.E. 17	U.PISA	
	INTRODUZIONE ALL'APL/360		
R -21 -6	C.N.U.C.E. 18	U.PISA	
	INTRODUZIONE AL CP - 67/GMS		
RA -21 -1	NASLIN P.	DUNOD	58
	PRINCIPES DES CALCULATRICES NUMERIQUES AUTOMATIQUES		
R -21 -2	VARNER W.W.	RINEHART	
	COMPUTING WITH DESK CALCULATORS		
R -21 -19	CRAWFORD E.R.	PRENTICE-HALL	68
	INTRODUCTION TO DATA PROCESSING		
-21 -23	FLORE I.	PRENTICE-HALL	69
	COMPUTER ORGANIZATION		
-21 -251	GNEDENKO B.V.-KOROLIOUK W.S. IOUCHTCHENKO E.L.	DUNOD	69
	ELEMENTS DE PROGRAMMATION SUR ORDINATEURS		
-21 -15	IRWIN W.C.	VAN NOSTRAND	60
	DIGITAL COMPUTER PRINCIPLES		
-21 -12	MOSES M.	GERNSBACH	59
	PRINTED CIRCUITS		
-21 -14A	MURPHY J.	RIDER	58
	BASIC OF DIGITAL COMPUTERS - VOL.1		
-21 -16	VAN ALLEN R.L.-HOUSE B.C.	NAVAL R.L.	58
	NONDESTRUCTIVE READOUT OF MULTILEVEL MAGNETIC MEMORY		
-21 -13	WILKES M.V.-WHEELER D.J.-GILL S.	ADDISON WESLEY	57
	THE PREPARATION OF PROGRAMS FOR AN ELECTRONIC DIGITAL COMPUTER		
-21 -21	WILKES M.V.	MAC DONALD	69
	TIME - SHARING COMPUTER SYSTEMS		

22 LINGUAGGI DI PROGRAMMAZIONE

R -22 -4	LAUTIER D.-LERNER J.P. COURS DE BASIC - ANALYSE ET PROGRAMMATION	MASSON	73
R -22 -3	MAURER W.D. THE PROGRAMMER'S INTRODUCTION TO LISP	MACDONALD	72
R -22 -2	RALSTON A. FORTRAN IV PROGRAMMING. A CONCISE EXPOSITION	MCGRAW-HILL	71
-22 -15	AMBROSIO S. LINGUAGGI ALGEBRICI	BORINGHIERI	65
-22 -14	BOLT A.B.-WARDLE M.E. COMMUNICATING WITH A COMPUTER	U.CAMBRIGE	70
-22 -12	ERSHOV A.P. PROGRAMMING PROGRAMME FOR THE BESM COMPUTER	PERGAMON PRESS	59
-22 -5	GOODMAN ANNUAL REVIEW AUTOMATIC PROGRAMMING	PERGAMON PRESS	60
-22 -8	LAFITTE R. INITIATION AU LANGAGE SCHOOL	DUNOD	69
-22 -7	MC KRACKEN D. A GUIDE TO FORTRAN PROGRAMMING	WILEY J.-SONS	61
-22 -13	NOLIN L. FORMALISATION DES NOTIONS DE MACHINE ET DE PROGRAMME	GAUTHIER-VILLARS	69
-22 -16	PETERSON ERROR - CORRECTING CODES	WILEY J.-SONS	61
-22 -11	REVIGLIO G. I LINGUAGGI DEGLI ELABORATORI ELETTRONICI	BORINGHIERI	64
-22 -9	VALSESIA S. EVOLUZIONE DEI LINGUAGGI DEGLI ELABORATORI ELETTRONICI	BORINGHIERI	65

31 CALCOLO DELLE PROBABILITA'

-310-61	ARTHURS A.M. PROBABILITY THEORY	ROUTLEDGE-KEGAN	67
-310-75	DABONI L. ELEMENTI DI CALCOLO DELLE PROBABILITA'	BORINGHIERI	67
RA -310-32	POMPILJ G. COMPLEMENTI DI CALCOLO DELLE PROBABILITA'	EREDI V. VESCHI	49

RA -311-46	BERTRAND J. CALCUL DES PROBABILITE'	GAUTHIERS VILLARS	07
R -313-1A	BOCHNER S. HARMONIC ANALYSIS AND THE THEORY OF PROBABILITY	U.CALIFORNIA PRESS	55
RA -314-29	AVONDO BODINO G. I PROCESSI STOCASTICI IN STATISTICA	BOCCONI	60
-314-64	COX D.R.-LEWIS P.A.W. L'ANALYSE STATISTIQUE DES SERIES D'EVENEMENTS	DUNOD	69
RA -314-10	KAI LAY CHUNG MARKOV CHAINS WITH STATIONARY TRANSITION PROBABILITIES	SPRINGER VERLAG	60
-315-24	MORGAN B.W. AN INTRODUCCION TO BAYESIAN STATISTICAL DECISION PROCESSES	PRENTICE HALL	68
RA -315-6	BROSS I.D.F. DESIGN FOR DECISION	MACMILLAN CO.	53
-316-5	COX D.R.-SMITH W.L. QUEUES	METHUEN CO.	68
-316-12	TOSALLI A. ELEMENTI INTRODUTTIVI ALLA TEORIA DELLE CODE	BORINGHIERI	68
R -317-2	BAZOVSKY J. RELIABILITY THEORY AND PRACTICE	PRENTICE HALL	61
R -318-1	ASH R. INFORMATION THEORY	INTERSCIENCE PUBL.	67
RA -319-12	FERRERI C. CRITERI DI ANALISI DELLA DINAMICA DEGLI AGGREGATI DEL CONTO DELLA PRODUZIONE	ABBACO	68
-319-31	WEAVER W. THE THEORY OF PROBABILITY	HEINEMANN	65
RA -319-14	PREDETTI A. CARATTERISTICHE DI UNO SCHEMA GENERALE DI PROVE RIPETUTE CON PROB. DIPENDENTI	ABETE	63
-319-	DORE P. INTROD.AL CALCOLO DELLE PROB. E ALLE SUE APPLICAZIONI INGEGNERISTICHE	PATRON	62
RA -319-16	WOODWARD P.M. PROBABILITIES ANALYSE FREQUENTIELLE, INFORMATION THEORIE DU RADAR	EYROLLES	60
32 DISTRIBUZIONI CAMPIONARIE *****			
R -32 -1	AMATO V. FORME QUADRATICHE IN STATISTICA	GIUFFRE'DOTT.A.	58

RA -32 -6	ELDERTON W.P.-JOHNSON N.L.	V.CAMBRIDGE PRESS	69
	SYSTEMS OF FREQUENCY CURVES		
R -32 -3	KENDALL M.G.	GRIFFIN	72
	A COURSE IN MULTIVARIATE ANALYSIS, BEING NUMBER TWO		
R -32 -5	MARDIA K.V.	GRIFFIN	70
	FAMILIES OF BIVARIATE DISTRIBUTIONS, BEING NUMBER TWENTY-SEVEN		
R -32 -2	NEWMAN T.G.-ODELL P.L.	GRIFFIN	71
	THE GENERATION OF RANDOM VARIATES, BEING NUMBER TWENTY-NINE		
R -32 -4	ORD J.K.	GRIFFIN	72
	FAMILIES OF FREQUENCY DISTRIBUTIONS, BEING NUMBER THIRTY		
RA -32 -7	STICHPROBEN	STAT. BUNDESAMT	60
	...IN DER AMTLICHEN STATISTIK		
-32 -8	AITCHISON J.-BROWN J.A.C.	V.CAMBRIDGE PRESS	57
	THE LOGNORMAL DISTRIBUTION		
-32 -9	COCHRAN W.	WILEY J. - SONS	53
	SAMPLING TECHNIQUES		

.33 STIMA

R -33 -10	DEMING W.E.	WILEY J.-SONS	50
	SOME THEORY OF SAMPLING		
RA -33 -5	DEUTSCH R.	PRENTICE HALL INC.	65
	ESTIMATION THEORY		
RA -33 -13	DUMAS DE RAULY D.	GAUTHIER-VILLARS	66
	L'ESTIMATION STATISTIQUE		
RA -33 -16	LINNIK Y.V.	PERGAMON PRESS	61
	METHOD OF LEAST SQUARES AND PRINCIPLES OF THE THEORY OF OBSERVATIONS		
RA -33 -19	OVE F.	FOA REPRO	71
	STATISTICAL INFERENCE IN GRAPHS		
R -33 -9	VAJANI L.	O.L.C.	65
	METODO DEL CAMPIONE E RICERCHE DI MERCATO		
R -33 -1	WASAN M.T.	MCGRAW-HILL BOOK C.	70
	PARAMETRIC ESTIMATION		
RA -33 -11	RAO C.R.	WILEY J.-SONS	52
	ADVANCED STATISTICAL METHODS IN BIOMETRIC RESEARCH		
R -33 -20	ANDREWS D.F.-BICKEL P.J.-HAMPEL F.R.	PRINCETON UN. PRESS	72
	ROBUST ESTIMATES OF LOCATION		

R -33 -2	BLOM G.	WILEY J.-SONS	58
STATISTICAL ESTIMATES AND TRANSFORMED BETA-VARIABLES			
R -33 -4	BRAMBILLA F.	GOLIARDICA	56
STATISTICA VOL.2			
R -33 -15	DURBING J.	LONDON	59
EFFICIENT ESTIMATION OF PARAMETERS IN MOVING-AVERAGE MODELS			
R -33 -3	GOOD I.J.	M.I.T PRESS	65
THE ESTIMATION OF PROBABILITIES AN ESSAY ON MODERN BAYESIAN METHODS			
34 VERIFICA DELLE IPOTESI			

R -34 -19	BRADLEY J.V.	PRENTICE HALL	68
DISTRIBUTION FREE STATISTICAL TESTS			
R -34 -9	CUCCONI O.	CEDAM	68
NUOVI METODI STATISTICI DI STIMA E DI VERIFICA D'IPOTESI			
R -34 -26	DAVID H.A.	WILEY J.-SONS	70
ORDER STATISTICS			
R -34 -22	EKLUND G.	ALMQVIST-WIKSELLS	60
STUDIES OF SELECTION BIAS IN APPLIED STATISTICS			
R -34 -23	FREUNF J.E. LIVERMORE P.E. MILLER I.	PRENTICE HALL	60
MANUAL OF EXPERIMENTAL STATISTICS			
R -34 -25	GOSH B.K.	ADDISON-WESLEY	70
SEQUENTIAL TEST OF STATISTICAL HYPOTHESES			
R -34 -31	GIARDINA B.	ANGELI F.	72
STATISTICA NON PARAMETRICA			
R -34 -4	HAJEK J.	CZECHOSLOVAK A.OFS.	67
THEORY OF RANK TESTS			
R -34 -28	HALL W.J.	N.C.I.S	66
SEQUENTIAL EVALUATION OF GRADED PREFERENCES FOR TWO TREATMENTS			
R -34 -7	HUITSON A.	GRIFFIN C.CO.	66
THE ANALYSIS OF VARIANCE			
R -34 -27	KINGSTON J.	S.G.I.B	45
A TEORIA DA INDUCAO ESTATISTICS			
R -34 -18	LI J.C.R.	ANN ARBOR	57
INTRODUCTION TO STATISTICAL INFERENCE			
R -34 -16	MANGANO G.	VITA E PENS.	70
SU ALCUNE PROPRIETA' DI UN TEST IMPIEGABILE NELL'ANALISI DELLA VARIANZA			

R -34 -5	NADDEO A. CONTRIBUTI ALLA TEORIA STATISTICA DEI CAMPIONI	GIUFFRE' A.	60
R -34 -8	NADDEO A. LA TEORIA DEI TEST STATISTICI	GIUFFRE' A.	63
R -34 -18	PLACKETT R.L. PRINCIPLES OF REGRESSION ANALYSIS	KLARENDON PRESS	60
R -34 -3	PURI L.M. NON PARAMETRIC TECHNIQUES IN STATISTICAL INFERENCE	U.PRESS	70
R -34 -17	ROY S.N. SOME ASPECTS OF MULTIVARIATE ANALYSIS	WILEY J.SONS	57
R -34 -32	ZACKS SELEMYAHU THE THEORY OF STATISTICAL INFERENCE	WILEY J.SONS	71
R -34 -11	ZANELLA A. SULLA FUNZIONE DI POTENZA DEL TEST T MULTIPLO	VITA E PENS.	71
25 METODI DI STATISTICA DESCRITTIVA *****			
R -35 -13	BARBERI B. RILEVAZIONI STATISTICHE	EINAUDI	57
R -35 -15	BARBERI B. IL METODO STATISTICO NELLO STUDIO DI FENOMENI OSSERVAZIONALI	BORINGHIERI	72
R -35 -8	BROWN R.G. SMOOTHING, FORECASTING AND PREDICTION OF DISCRETE TIME SERIES	PRENTICE HALL	73
R -35 -28	CALDT G. COURS DE STATISTIQUES DESCRIPTIVE	DUNOD	65
R -35 -26	CASTELLANO V, ISTITUZIONI DI STATISTICA	ILARDI	62
R -35 -29	DAVIS H.T. THE ANALYSIS OF ECONOMIC TIME SERIES	PRINCIPIA PRESS	41
R -35 -56	DE VERGOTTINI M. MEDIE, VARIABILITA', RAPPORTI	EINAUDI	57
R -35 -7	GALVANI L. STATISTICA, METODOLOGICA-VOL.1-PARTE 2-	GIUFFRE' A.	34
R -35 -25	GINI C, CORSO DI STATISTICA	VESCHI	55
R -35 -9	GRENANDER U. ROSEMBLATT M. STATISTICAL ANALYSIS OF STATIONARY TIME SERIES	WILEY J.-SONS	57

R -35 -40	GUILBAND G.T. STATISTIQUES DES CRONIQUES	DUNOD	68
R -35 -12	HANNAN E.J. MULTIPLE TIME SERIES	WILEY J.-SONS	70
R -35 -17	MARSILI LIBELLI M. METODOLOGIA STATISTICA	BATTISTELLI	24
R -35 -11	QUENOUILLE M.H. THE ANALYSIS OF MULTIPLE TIME SERIES	HAFNER P.CO.	57
R -35 -1	QUINET E. SERIES TEMPORELLES ET DECISIONS ECONOMIQUES	DUNOD	69
R -35 -2	RICHARDS J.W. INTERPRETATION OF TECHNICAL DATA	D.VAN N.CO.	67
R -35 -31	SENDERS V.L. MEASUREMENT AND STATISTIC- A BASIC TEST	OXFORD U.P.	58
R -35 -26	VAJANI L. ELEMETNTI DELLA TEORIA DELLE SERIE TEMPORALI	VITA E PENS.	69
R -35 -51	VINCI F. INTRODUZIONE AL METODO STATISITCO	CEDAM	30
R -35 -37	WOLBELG J.R. PREDICTION ANALYSIS	D.VAN N.CO.	67
.26 METODI DI CAMPIONAMENTO			

R -36 -21	CASTELLANO V.-HERZEL A. ELEMENTI DI TEORIA DEI CAMPIONI	ILARDI	71
R -36 -4	COCHRAN W.G. SAMPLING TECHNIQUES	WILEY J.-SONS	63
R -36 -2	DEMING W.E. SAMPLE DESIGN IN BUSINESS RESEARCH	WILEY J.-SONS	63
R -36 -8	DESABIE J. THEORIE ET PRATIQUE DES SONDAGES	DUNOD	66
R -36 -17	LAWLEY D.N.-MAXWELL A.E. FACTOR ANALYSIS AS A STATISTICAL METHOD	BUTTERWORTHS	71
R -36 -9	NICEFORO A. IL METODO STATISTICO-TEORIA E APPLICAZIONI ALLE SCIENZE NATURALI, SOCIALI E ARTE	PRINCIPATO	23
R -36 -16	RAY D. SAMPLING THEORY	MC.GRAW HILL	68

R -36 -15	STUART A.	C.GRIFFIN CO.	62
BASIC IDEAS OF SCIENTIFIC SAMPLING			
R -36 -12	JAMARNE T.	PRENTICE HALL	67
ELEMENTARY SAMPLING THEORY			
R -36 -3	YATES F.	C.GRIFFIN CO.	60
SAMPLING METHODS FOR CENSUSES AND SURVEYS			
R -36 -1	DALENIUS T.	ALMQUIST W.	57
SAMPLING IN SWEDEN. CONTRIBUTIONS TO THE METHODS AND THEORIES OF SAMPLE SURVEY			
R -36 -5	HENDRICKS W.A.	SCARECRON PRESS	56
THE MATHEMATICAL THEORY OF SAMPLING			
R -36 -14	JOHNSON P.O.-RAO M.S.	U.M.PRESS	59
MODERN SAMPLING METHODS-THEORY, EXPERIMENTATION, APPLICATION			
R -36 -7	SAMFORD M.R.	OLIVER AND BOYD	62
IN INTRODUCTION TO SAMPLING THEORY. WITH APPLICATIONS TO AGRICULTURE			
R -36 -10	SCHWARTZ D.-LAZAR P.	EMF	64
ELEMENTS DE STATISTIQUE MEDICALE ET BIOLOGIQUE			
R -36 -11	SUKHATNE P.V.	JS	
SAMPLING THEORY OF SURVEYS WITH APPLICATIONS			
37 PIANO DEGLI ESPERIMENTI			

R -37 -51	BARNES R.M.	WILEY J.-SONS	57
WORK SAMPLING			
-37 -72	CHEW V.	WILEY J.-SONS	59
EXPERIMENTAL DESIGNS IN INDUSTRY			
R -37 -1	COCHRAN W.G.-COX G.M.	WILEY J.-SONS	57
EXPERIMENTAL DESIGNS			
-37 -68	DAVIES O.L.	OLIVER-BOYD	54
THE DESIGN AND ANALYSIS OF INDUSTRIAL EXPERIMENTS			
R -37 -2	COX D.R.	WILEY J.-SONS	58
PLANNING OF EXPERIMENTS			
R -37 -49	DAVID H.A.	GRIFFIN CO.	69
THE METHOD OF PAIRED COMPARISONS			
R -37 -10	DUGUE D.-GIRAULT M.	DUNOD	59
ANALYSE DE VARIANCE ET PLANS D'EXPERIENCE			
R -37 -19	ELDERTON W.P.	U.PRESS	53
FREQUENCY CORVES AND CORRELATION			

R -37 -5	FEDERER W.T. EXPERIMENTAL DESIGN. THEORY AND APPLICATION	MACMILLAN	55
R -37 -4	FISHER R.A. LA PROGRAMMAZIONE DEGLI ESPERIMENTI	NISTRJ-LISCHI	54
R -37 -17	FRYER H.C. CONCEPTS AND METHODS OF EXPERIMENTAL STATISTICS	ALLYN-BACON	66
R -37 -38	GUENTHER W.C. ANALYSIS OF VARIANCE	PRENTICE HALL	64
R -37 -20	HOPE K. METHODS OF MULTIVARIATE ANALYSIS	LONDON PRESS	68
R -37 -46	JOHN P.W. STATISTICAL DESIGN AND ANALYSIS OF EXPERIMENTS	MACMILLAN	71
R -37 -41	KENDALL N.G.-STUART A. THE ADVANCED THEORY OF STATISTICS	GRIFFIN	66
R -37 -33	KEMPTHORNE O. THE DESIGN AND ANALYSIS OF EXPERIMENTS	WILEY J.-SONS	52
R -37 -27	ZANELLA A. UN NUOVO TEST PER L'ANALISI DEI MODELLI LINEARI	IGIS	66
-37 -66	FRUND J.E.-LIVERMORE P.E.-MILLER J. MANUAL OF EXPERIMENTAL STATISTICS	PRENTICE HALL	60
-37 -76	KENDALL M.G. A COURSE IN MULTIVARIATE ANALYSIS	HAWKINS	57
-37 -60	VAJDA S. THE MATHEMATICS OF EXPERIMENTAL DESIGN	GRIFFIN	67
28 RELAZIONI TRA VARIABILI *****			
-38 -25	ANDERSON T.W. INTRODUCTION TO MULTIVARIATE STATISTICAL ANALYSIS	WILEY J. SONS	58
R -38 -12	ACTON F.S. ANALYSIS OF STRAIGHT LINE DATA	WILEY J. SONS	59
R -38 -11	BRAMBILLA F. TEORIA STATISTICA DELLA CORRELAZIONE E DELLA CONNESSIONE	M. RICCI	42
R -38 -10	CUSIMANO G. LA METODOLOGIA STATISTICA CONDIZIONATA DELL'ANALISI DI PIU' VARIABILI	DELF	55
R -38 -9	GOLDBERGER A.S. TOPICS IN REGRESSION ANALYSIS	MACMILLAN	68

R -38 -1	GUITTON H. STATISTIQUE ET GEOMETRIE	DALLOZ	59
R -38 -2	LYLE P. REGRESSION ANALYSIS OF PRODUCTION COSTS AND FACTORY OPERATIONS	OLIVER-BOYD	57
R -38 -15	PURI M.L.-SEN P.K. NON PARAMETRIC METHODS IN MULTIVARIATE ANALYSIS	WILEY J.SONS	71
R -38 -8	QUENOUILLI M.H. ASSOCIATED MEASUREMENTS	BUTTERWORTHS	52
R -38 -3	SALVEMINI T. REGRESSIONE E CORRELAZIONE	EINAUDI	59
R -38 -13	SMILLIE K.W. AN INTRODUCCION TO REGRESSION AND CORRELATION	RYERSON	66
-38 -21	COOLEY W.W.-LOHNES P.R. MULTIVARIATE PROCEDURES FOR THE BEHAVIORAL SCIENCES	WILEY J.SONS	62
A -38 -28	CORRIDORE F. CORRELAZIONI STATISTICHE	MANUZIO	15
A -38 -37	CZUBER E. DE STATISTICHEN FORSCHUNGSMETHODEN	VERLAG-SEIDEL	38
-38 -36	DEMPSTER A.P. ELEMENTS OF CONTINUOUS MULTIVARIATE ANALYSIS	ADDISON-WESLEY	69
A -38 -22	DIEULEFAIT C.E. THEORIE DE LA CORRELATION	ROSARIO	35
-38 -25	FERGURSON T. ON THE EXISTENCE OF LINEAR STRUCTURAL	U.CALIFORNIA	55
-38 -24	SCARDOVI I. I FONDAMENTI EMPIRICI DELLA VIARJABILE STATISTICA DOPPIA	PATRON	70
-38 -23	SPRENT P. MODELS IN REGRESSION AND RELATED TOPICS	MZTHUEN CO.	69
R -38 -4	TSCHUPROW A.A. PRINCIPLES OF THE MATHEMATICAL THEORY OF CORRELATION	W.HODGE CO.	39

```

*****
*****
**
**
**
**      I N T E R P R E T A T O R E   D E L   L I N G U A G G I O
**
**      == L I N G U E B ==
**
**      P E R   L A   C O S T R U Z I O N E ,   G E S T I O N E   E   I N T E R R O G A Z I O N E
**
**      A U T O M A T I C A   D I   U N A   B I B L I O T E C A
**
**      I N   F O R M A   B A T C H
**
**
**
*****
*****

```

```

*      E L E N C O   D E I   C O M A N D I

*      /1/CTC. # COSTRUZIONE DEL TRADUTTORE DELLE PAROLE CHIAVE
          NELLE QUATTRO LINGUE

*      /1/CMC. # COSTRUZIONE DELLA MATRICE DI CLASSIFICAZIONE
          DELLE PAROLE CHIAVE

*      /1/CAL. # COSTRUZIONE DELL*ARCHIVIO DEI LIBRI

*      /2/ATC. # AGGIORNAMENTO DEL TRADUTTORE E DELLA MATRICE DI
          CLASSIFICAZIONE DELLE PAROLE CHIAVE

*      /2/ACA. # AGGIORNAMENTO DI UN ELEMENTO DELLA MATRICE DI
          CLASSIFICAZIONE DELLE PAROLE CHIAVE

*      /2/AAL. # AGGIORNAMENTO DELL*ARCHIVIO DEI LIBRI

*      /3/RUI. # RICERCA DELL*UTENTE

*      /3/SLL. # STAMPA DELLA LISTA DEI LIBRI DELLA CLASSE RICHIESTA,
          PRECEDENTEMENTE INDIVIDUATA

*      /S/ALT. # FINE DELL*ESECUZIONE

```

```

PROGRAM SALE(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE2)
INTEGER CMND,CTC,CMC,CAL,ATC,ACA,AAL,RUT,SLL,ALT
COMMON LIBRI(1)
COMMON/UNO/ MAT(250,8)
COMMON/ONE/ ALFA(26)
COMMON/DUE/IU
COMMON/TWO/ COD,IUG
COMMON/TRE/ SG(4),LG(4)
COMMON/THREE/EEA,EEB,EEC
COMMON/QUATTRO/ N,JND1,IND2,IND3,IND4
COMMON/SETTE/ ITEMP(8)
COMMON/ING/ EA(250),EB(250),EC(250),ED(250)
COMMON/ITA/ IA(250),IB(250),IC(250),ID(250)
COMMON/FRA/ FA(250),FB(250),FC(250),FD(250)
COMMON/IED/ DA(250),DB(250),DC(250),DD(250)
COMMON/LLT/ ELET(26),ILE1(26),FLET(26),DLET(26)
CTC=7R/1/CTC.
CMC=7R/1/CMC.
CAL=7R/1/CAL.
ATC=7R/2/ATC.
ACA=7R/2/ACA.
AAL=7R/2/AAL.
RUT=7R/3/RUT.
SLL=7R/3/SLL.
ALT=7R/S/ALT.
150 READ(5,1) CMND
WRITE(6,20)
IF(CMND.NE.CTC) GOTO 100
IF(IND1.EQ.7RMADECTC) GOTO 300
CALL CREA1
GOTO 150
100 IF(CMND.NE.CMC) GOTO 200
IF(IND1.NE.7RMADECTC) GOTO 250
IF(IND2.EQ.7RMADECMC) GOTO 350
CALL CREA2
GOTO 150
200 IF(CMND.NE.CAL) GOTO 400
IF(IND3.EQ.7RMADECAL) GOTO 450
CALL CREA3
GOTO 150
400 IF(CMND.NE.ATC) GOTO 500
IF(IND2.EQ.7RMADECMC) GOTO 410
IF(IND1.EQ.7RMADECTC) GOTO 550
GOTO 560
410 CALL AGG1
GOTO 150
500 IF(CMND.NE.ACA) GOTO 600
IF(IND2.EQ.7RMADECMC) GOTO 510
IF(IND1.EQ.7RMADECTC) GOTO 550
GOTO 560
510 CALL AGG2
GOTO 150
600 IF(CMND.NE.AAL) GOTO 700

```

PAUSE

PAUSE

PAUSE

PAUSE

PAUSE

```

IF(IND3.NE.7RMADECAL) GOTO 650
CALL AGG3
GOTO 150
700 IF(CMND.NE.RUT) GOTO 800
IF(IND3.EQ.7RMADECAL.AND.JND2.EQ.7RMADECMC) GOTO 710
GOTO 750
710 WRITE(6,19)
WRITE(6,11)
CALL UTENTE
GOTO 150
800 IF(CMND.NE.SLL) GOTO 900
IF(IND4.NE.7RMADERUT) GOTO 850
WRITE(6,22)
WRITE(6,11)
CALL STAMPA
GOTO 150
900 IF(CMND.NE.ALT) GOTO 1000
STOP
250 WRITE(6,2)
STOP
300 WRITE(6,3)
STOP
350 WRITE(6,4)
STOP
450 WRITE(6,5)
STOP
550 WRITE(6,6)
STOP
560 WRITE(6,7)
STOP
650 WRITE(6,8)
STOP
750 WRITE(6,9)
STOP
850 WRITE(6,21)
STOP
1000 WRITE(6,10)
1 FORMAT(R7)
2 FORMAT(1X,*VVV FASE NON ESEGUIBILE PER MANCANZA*,
** ARCHIVIO CHIAVI VVV*)
3 FORMAT(1X,*VVV ATTENZIONE ARCHIVIO CHIAVI GIA *,
**ESISTENTE VVV*)
4 FORMAT(1X,*VVV ATTENZIONE MATRICE CLASSIFICAZIONI GIA *,
**ESISTENTE VVV*)
5 FORMAT(1X,*VVV ATTENZIONE ARCHIVIO LIBRI GIA *,
**ESISTENTE VVV*)
6 FORMAT(1X,*VVV AGGIORNAMENTO NON ESEGUIBILE PER MANCANZA*,
** MATRICE CLASSIFICAZIONE VVV*)
7 FORMAT(1X,*VVV AGGIORNAMENTO NON ESEGUIBILE PER MANCANZA*,
** ARCHIVIO CHIAVI E MATRICE CLASSIFICAZIONE VVV*)
8 FORMAT(1X,*VVV AGGIORNAMENTO NON ESEGUIBILE PER MANCANZA*,
** ARCHIVIO LIBRI VVV*)
9 FORMAT(1X,*VVV ARCHIVIO NON UTILIZZABILE VVV*)

```

```
10  FORMAT(1X,*VVV  COMANDO ERRATO  VVV*)
11  FORMAT(///)
19  FORMAT(1X,*R I C E R C A   D E L L   U T E N T E *)
20  FORMAT(1H1)
21  FORMAT(1X,*VVV  ATTENZIONE NON E STATO DATO IL COMANDO PER *,
**INDIVIDUARE LE CLASSI  VVV*)
22  FORMAT(1X,*LISTA DEI LIBRI DELLA CLASSE RICHIESTA*)
    STOP
    END
```

SUBROUTINE CREAL

```

*
*
*   COSTRUISCE IL TRADUTTORE DELLE CHIAVI NELLE QUATTRO LINGUE
*   INGLESE, ITALIANO, FRANCESE, TEDESCO
*
*
*.....N = NUMERO DELLE PAROLE CHIAVE
*.....EA,EB,EC = INSIEME DI VETTORI CONTENENTI LA LISTA DELLE PAROLE
*   CHIAVE NELLA LINGUA INGLESE
*.....IA,IB,IC = INSIEME DI VETTORI CONTENENTI LA LISTA DELLE PAROLE
*   CHIAVE NELLA LINGUA ITALIANA
*.....FA,FB,FC,
*.....FA,FB,FC = INSIEME DI VETTORI CONTENENTI LA LISTA DELLE PAROLE
*   CHIAVE NELLA LINGUA FRANCESE
*.....DA,DB,DC = INSIEME DI VETTORI CONTENENTI LA LISTA DELLE PAROLE
*   CHIAVE NELLA LINGUA TEDESCA
*.....ALFA = VETTORE DELLE LETTERE DELL*ALFABETO
*.....ELET, ILET, FLET, DLET = VETTORI DEI PUNTATORI AI RISPETTIVI
*   GRUPPI DI PAROLE IN ORDINE ALFABETICO
*.....PI, PF, PD = PUNTATORI DELLE CHIAVI DALLA LINGUA INGLESE RI-
*   SPETTIVAMENTE ALLA LINGUA ITALIANA, FRANCESE, TEDESCA
*.....ID, FD, DD = PUNTATORI RISPETTIVAMENTE DALLA LINGUA ITALIANA,
*   FRANCESE, TEDESCA ALLA LINGUA INGLESE
*.....IND1 = INDICATORE DELL*AVVENUTA COSTRUZIONE DEL TRADUTTORE
*   DELLE CHIAVI
*
*
*   INTEGER ALFA
*   INTEGER T1,T2,T3
*   INTEGER SG,COD
*   INTEGER EEA,EEB,EEC
*   INTEGER ELET,FLET,DLET
*   INTEGER EA,EB,EC,ED,FA,FB,FC,FD,DA,DB,DC,DD,PI,PF,PD
*   COMMON/UNO/ MAT(250,8)
*   COMMON/ONE/ ALFA(26)
*   COMMON/TWO/ IU
*   COMMON/TWO/ COD,JUG
*   COMMON/TRE/ SG(4),LG(4)
*   COMMON/THREE/EEA,EEB,EEC
*   COMMON/QUATTRO/ N,IND1,IND2,IND3,IND4
*   COMMON/SETTE/ ITEMP(8)
*   COMMON/ING/ EA(250),EB(250),EC(250),ED(250)
*   COMMON/ITA/ IA(250),IB(250),IC(250),ID(250)
*   COMMON/FRA/ FA(250),FB(250),FC(250),FD(250)
*   COMMON/TED/ DA(250),DB(250),DC(250),DD(250)
*   COMMON/PUNT/ PI(250),PF(250),PD(250)
*   COMMON/LLT/ ELET(26),ILET(26),FLET(26),DLET(26)
*   COMMON/TTE/ T1(250),T2(250),T3(250)
*   DATA (ALFA(I),I=1,26)/01B,02B,03B,04B,05B,06B,07B,10B,11B,12B,13B,
*14B,15B,16B,17B,20B,21B,22B,23B,24B,25B,26B,27B,30B,31B,32B/
*   READ(5,11) N
*   READ(5,1) (EA(I),EB(I),EC(I),I=1,N)

```

```
READ(5,1) (IA(I),JB(I),IC(I),J=1,N)
READ(5,1) (FA(I),FB(I),FC(I),J=1,N)
READ(5,1) (DA(I),DB(I),DC(I),J=1,N)
DO 100 I=1,N
ED(I)=ID(I)=FD(I)=DD(I)=I
100 CONTINUE
CALL ORDIN (EC,N,EA,EB,ED)
CALL ORDIN (IC,N,IA,IB,ID)
CALL ORDIN (FC,N,FA,FB,FD)
CALL ORDIN (DC,N,DA,DB,DD)
DO 200 I=1,N
CALL PUNT (ED,PI,ID,N,I)
CALL PUNT (ED,PF,FD,N,I)
CALL PUNT (ED,PD,DD,N,I)
200 CONTINUE
DO 300 I=1,N
CALL PUNT (ID,T1,ED,N,I)
CALL PUNT (FD,T2,ED,N,I)
CALL PUNT (DD,T3,ED,N,I)
300 CONTINUE
DO 400 I=1,N
ID(I)=T1(I)
FD(I)=T2(I)
DD(I)=T3(I)
400 CONTINUE
CALL FWORD (EA,N,ELET)
CALL FWORD (IA,N,ILET)
CALL FWORD (FA,N,FLET)
CALL FWORD (DA,N,DLET)
IND1=7RMADECTC
1 FORMAT(3A10)
11 FORMAT(15)
RETURN
END
```

```

SUBROUTINE CREA2
*
*
*   COSTRUISCE LA MATRICE DI CLASSIFICAZIONE DELLE CHIAVI
*
*.....MAT = MATRICE DI CLASSIFICAZIONE DELLE CHIAVI
*.....IVET = VETTORE DI CONTROLLO DELLA CORRISPONDENZA FRA LE RIGHE
*           DELLA MAT E LE RISPETTIVE CHIAVI
*.....IND2 = INDICATORE DELL*AVVENUTA COSTRUZIONE DELLA MAT
*
*
*   DIMENSION IVET(250)
*   INTEGER EA,EB,EC
*   COMMON/UNO/ MAT(250,8)
*   COMMON/QUATTRO/ N,IND1,IND2,IND3,IND4
*   COMMON/ING/ EA(250),EB(250),EC(250),ED(250)
*   DO 100 I=1,N
*   READ(5,1) (MAT(I,J),J=1,8),IVET(I)
*   IF(IVET(I).NE.I) GO TO 200
100  CONTINUE
*   IND2=7RMADECMC
*   GO TO 300
200  WRITE(6,3)
*   STOP
*                                     PAUSE
1    FORMAT(7A10,A3,I7)
3    FORMAT(1X,*VVV INPUT ERROR - ORDINE ERRATO NELLA CLASSIFICAZIONE
* VVV*)
200  RETURN
*   END

```

```

SUBROUTINE CREA3
*
*
*   COSTRUISCE L*ARCHIVIO DEI LIBRI
*
*.....LIBRI = VETTORE CONTENENTE L*ARCHIVIO DEI LIBRI
*.....MPUNT = VETTORE DEI PUNTATORI AGLI ESTREMI SUPERIORI
*   DELLE CLASSI
*.....NUOV = NUMERO DELLE CLASSI IN CUI SI ARCHIVIANO I LIBRI
*.....NCLASSE = NUMERO DELLA CLASSE NELLA QUALE SI ARCHIVIANO DEI LIBRI
*.....NSCHEDE = NUMERO DELLE SCHEDE CONTENENTI LE INFORMAZIONI DA
*   ARCHIVIARE
*.....IND3 = INDICATORE DELL*AVVENUTA COSTRUZIONE DELL*ARCHIVIO DEI
*   LIBRI
*
*
*   DIMENSION IDEN(8),NCAT(8)
*   COMMON LIBRI(1)
*   COMMON/QUATTRO/ N,IND1,IND2,IND3,IND4
*   COMMON/NOVE/ MPUNT(73)
*   KK=0
*   DO 100 I=1,73
*   IF(I.EQ.1) MPUNT(I)=8
*   IF(I.NE.1) MPUNT(I)=MPUNT(I-1)+8
*   READ(5,2) (IDEN(JI),JI=1,8)
*   DO 200 J=1,8
*   KK=KK+1
*   LIBRI(KK)=IDEN(J)
200  CONTINUE
100  CONTINUE
*
*
*   ENTRY AGG3
*
*
*   AGGIORNA L*ARCHIVIO DEI LIBRI
*
*
*   READ(5,3) NUOV
*   IF(NUOV.EQ.0) STOP
*   DO 300 LL=1,NUOV
*   READ(5,4) NCLASSE,NSCHEDE
*   NPOSTI=NSCHEDE*8
*   REWIND2
*   MMM=MPUNT(73)
*   LA=MPUNT(NCLASSE)
*   LD=LA+1
*   DO 500 LB=LD,MMM
500  CONTINUE
*   DO 600 I=1,NSCHEDE
*   READ(5,2) (NCAT(LC),LC=1,8)
*   DO 800 LE=1,8

```

PAUSE

```
LIBRI(LD)=NCAT(LE)
LD=LD+1
800 CONTINUE
600 CONTINUE
REWIND2
MMM=MMM+NPQSTI
DO 950 II=LD,MMM
READ(2,5) LIBRI(II)
950 CONTINUE
DO 1000 LE=NCLASSE,73
1000 MPUNT(LE)=MPUNT(LE)+NPOSTI
300 CONTINUE
IND3=7RMADECAL
2 FORMAT(8A10)
3 FORMAT(I5)
4 FORMAT(2I5)
5 FORMAT(A10)
RETURN
END
```

SUBROUTINE AGGI

```

*
*
*.....AGGIORNA IL TRADUTTORE E LA MATRICE DI CLASSIFICAZIONE DELLE CHIAVI
*
*
INTEGER COD
COMMON/UNO/ MAT(250,8)
COMMON/ONE/ ALFA(26)
COMMON/DUE/ IU
COMMON/TWO/ COD,IUG
COMMON/TRE/ SG(4),LG(4)
COMMON/THREE/EEA,EEB,EEC
COMMON/QUATTRO/ N,IND1,IND2,IND3,IND4
COMMON/SETTE/ ITEMP(8)
COMMON/ING/ EA(250),EB(250),EC(250),ED(250)
COMMON/ITA/ IA(250),IB(250),IC(250),ID(250)
COMMON/FRA/ FA(250),FB(250),FC(250),FD(250)
COMMON/TED/ DA(250),DB(250),DC(250),DD(250)
COMMON/LLT/ ELET(26),ILET(26),FLET(26),DLET(26)
COMMON/PUNT/ PI(250),PF(250),PD(250)
NS=N+1
READ(5,21) SG(1),LG(1),EA(NS),EB(NS),EC(NS)
READ(5,21) SG(2),LG(2),IA(NS),IB(NS),IC(NS)
READ(5,21) SG(3),LG(3),FA(NS),FB(NS),FC(NS)
READ(5,21) SG(4),LG(4),DA(NS),DB(NS),DC(NS)
CALL TEST1
IUG=0
IU=0
CALL TEST2 (EA,EB,EC,ELET,N,KE1,KE2,KE3,LE1,IQ,JROE)
CALL TEST2 (IA,IB,IC,ILET,N,KJ1,KJ2,KJ3,LI1,IQ,JROI)
CALL TEST2 (FA,FB,FC,FLET,N,KF1,KF2,KF3,LF1,IQ,JROF)
CALL TEST2 (DA,DB,DC,DLET,N,KD1,KD2,KD3,LD1,IQ,JROD)
PI(IQ)=PF(IQ)=PD(IQ)=ID(IQ)=FD(IQ)=DD(IQ)=IQ
CALL SNOOPY (EA,EB,EC,ELET,N,KE1,KE2,KE3,LE1,IQ,JROE)
CALL SNOOPY (IA,IB,IC,ILET,N,KJ1,KJ2,KJ3,LI1,IQ,JROI)
CALL SNOOPY (FA,FB,FC,FLET,N,KF1,KF2,KF3,LF1,IQ,JROF)
CALL SNOOPY (DA,DB,DC,DLET,N,KD1,KD2,KD3,LD1,IQ,JROD)
IF(COD.EQ.1) GOTO 900
IMAX=N-1
GOTO 1000
900 IMAX=N+1
1000 N=IMAX
CALL MODPUNT (ID,PI,N,LE1,LI1)
CALL MODPUNT (FD,PF,N,LE1,LF1)
CALL MODPUNT (DD,PD,N,LE1,LD1)
CALL RIGA (N,LE1)
21 FORMAT (2I1,3A10)
RETURN
END

```

SUBROUTINE AGG2

```

*
*
*.....AGGIORNA UN ELEMENTO DELLA MATRICE DJ CLASSIFICAZIONE DELLE CHIAVI
*
*
INTEGER COD
INTEGER ELET,FLET,DLET
COMMON/UNO/ MAT(250,8)
COMMON/TWO/ COD,IUG
COMMON/THREE/ EEA,EEB,EEC
COMMON/QUATTRO/ N,IND1,IND2,IND3,IND4
COMMON/ING/ EA(250),EB(250),EC(250),ED(250)
COMMON/LLT/ ELET(26),ILET(26),FLET(26),DLET(26)
READ(5,30) EEA,EEB,EEC
COD=6
CALL TEST3(EA,EB,EC,ELET,N,LE1)
500 READ(5,31) COD,NP,NL,IS
NPL=NP*10+NL
IF(NPL.GT.73) GOTO 750
CALL AGCLAS (NP,NL,LE1)
IF(IS.NE.57B) GO TO 500
DO 100 I=1,N
100 CONTINUE
GOTO 600
750 WRITE(6,33)
STOP
30 FORMAT(3A10)
31 FORMAT(3I1,R1)
33 FORMAT(UX,*VVV INPUT ERROR - AGGIORNAMENTO SU CLASSE*,
** NON ESISTENTE VVV*)
600 RETURN
END

```

PAUSE

SUBROUTINE UTENTE

```

*
*
* ESEGUE LA RICERCA DELL*UTENTE
*
*.....SIMILAR = VETTORE USATO PER DISCRIMINARE LA CLASSE/I RELATIVA
* ALLE PAROLE CHIAVE INTRODOTTE DALL*UTENTE
*.....IND4 = INDICATORE DELL*AVVENUTA RICERCA DELL*UTENTE
*
*

```

```

INTEGER COD,FD,DD
INTEGER SIMILAR
COMMON LIBRI(1)
COMMON/UNO/ MAT(250,8)
COMMON/TWO/ COD,JUG
COMMON/THREE/ EEA,EEB,EEC
COMMON/QUATTRO/ N,IND1,IND2,IND3,IND4
COMMON/SEI/ LP
COMMON/OTTO/ NCLA
COMMON/NOVE/ MPUNT(73)
COMMON/ING/ EA(250),EB(250),EC(250),ED(250)
COMMON/ITA/ IA(250),IB(250),IC(250),ID(250)
COMMON/FRA/ FA(250),FB(250),FC(250),FD(250)
COMMON/IED/ DA(250),DB(250),DC(250),DD(250)
COMMON/CCL/ ICLASS(20),SIMILAR(73),INDCL(73)
COMMON/LLT/ ELET(26),ILET(26),FLET(26),DLET(26)
NKEY=0
COD=7
READ(5,40) LLG,LB
IF(LB.NE.55B) GO TO 4000
IF(LLG.EQ.05B)GOTO 4100
IF(LLG.EQ.11B) GOTO 4200
IF(LLG.EQ.06B) GOTO 4300
IF(LLG.EQ.04B) GOTO 4400
4000 WRITE(6,41)
STOP
4100 CALL TEST3(EA,EB,EC,ELET,N,LE1)
NKEY=NKEY+1
IF(NKEY.GT.10) GO TO 4110
ICLASS(NKEY)=LE1
IF(LP.NE.57B) GOTO 4100
GOTO 4900
4200 CALL TEST3(IA,IB,IC,ILET,N,LI1)
NKEY=NKEY+1
IF(NKEY.GT.10) GO TO 4110
ICLASS(NKEY)=ID(LI1)
IF(LP.NE.57B) GOTO4200
GOTO 4900
4300 CALL TEST3(FA,FB,FC,FLET,N,LF1)
NKEY=NKEY+1
IF(NKEY.GT.10) GO TO 4110
ICLASS(NKEY)=FD(LF1)

```

PAUSE

```

      IF(LP.NE.57B) GOTO 4300
      GOTO 4900
4400  CALL TEST3(DA,DB,DC,DLET,N,LD)
      NKEY=NKEY+1
      IF(NKEY.GT.10) GO TO 4110
      ICLASS(NKEY)=DD(LD)
      IF(LP.NE.57B) GOTO 4400
      GOTO 4900
4110  WRITE(6,43)
      STOP
4900  WRITE(6,42)
      DO 6050 I=1,73
6050  SIMILAR(I)=0
      MASK=77B
      DO 6000 I=1,NKEY
      K=0
      IIR=0
      DO 6100 J=1,8
      JJ=1
      IT=ICLASS(I)
      MTEMP=MAT(IT,J)
6200  IF(IIR.EQ.1) GO TO 6350
      IF(J.EQ.8) JJ=8
      IF(J.EQ.8.AND.JJ.EQ.8) IIR=1
6350  MTEMP=(MTEMP*8)*8
      IK=MTEMP.AND.MASK
      K=K+1
      IF(IK.EQ.33B) GOTO 6300
      SIMILAR(K)=SIMILAR(K)+1
6300  JJ=JJ+1
      IF(JJ.EQ.11) GOTO 6100
      GOTO 6200
6100  CONTINUE
6000  CONTINUE
      MAX=0
      DO 7000 II=1,73
7000  MAX=MAX0(MAX,SIMILAR(II))
      NCLA=0
      DO 7100 II=1,73
      IF(SIMILAR(II).NE.MAX) GOTO 7100
      NCLA=NCLA+1
      INDCL(NCLA)=II
7100  CONTINUE
      IF(NCLA.NE.1) GOTO 8000
      WRITE(6,50)
      GOTO 8001
8000  WRITE(6,51)
8001  DO 8002 JJ=1,NCLA
      IF(INDCL(JJ).EQ.1) GOTO 8003
      IPREC=INDCL(JJ)-1
      LM1=MPUNT(IPREC)+1
      GOTO 8004
8003  LM1=1

```

PAUSE

```
8004  LM2=LM1+7
      WRITE(6,52) (LIBRI(J),I=LM1,LM2),INDCL(JJ)
8002  CONTINUE
      IND4=7RMADERUT
40    FORMAT(2R1)
41    FORMAT(1X,*VVV INPUT ERROR - CODICE DELLA LINGUA ERRATO VVV*)
42    FORMAT(///)
43    FORMAT(1X,*VVV INPUT ERROR - CHIAVI IN NUMERO *,
**SUPERIORE AL CONSENTITO VVV*)
50    FORMAT(1X,*C L A S S E I N D I V I D U A T A *,//)
51    FORMAT(1X,*VVV CHIAVI NON SELETTIVE VVV*,//,1X,
**C L A S S I I N D I V I D U A T E*,//)
52    FORMAT(1X,8A10,3X,*NUMERO PER STAMPA LISTA LIBRI*,2X,12)
      RETURN
      END
```

SUBROUTINE STAMPA

*
*
*
*
*

*.....STAMPA DELLA LISTA DEI LIBRI DELLA CLASSE RICHIESTA

```

COMMON LIBRI(1)
COMMON/QUATTRO/ N,IND1,IND2,IND3,IND4
COMMON/OTTO/ NCLA
COMMON/NOVE/ MPUNT(73)
COMMON/CCL/ ICLASS(20),SIMILAR(73),INDCL(73)
READ(5,1) KCLAS
DO 100 I=1,NCLA
100 IF(INDCL(I).EQ.KCLAS) GO TO 200
WRITE(6,2)
STOP
200 IF(KCLAS.EQ.1) GO TO 201
IPREC=KCLAS-1
LM1=MPUNT(IPREC)+1
GO TO 202
201 LM1=1
202 LM2=MPUNT(KCLAS)
WRITE(6,3) (LIBRI(I),I=LM1,LM2)
1 FORMAT(15)
2 FORMAT(1X,*VVV NUMERO DELLA CLASSE DA STAMPARE ERRATO VVV*)
3 FORMAT(1X,8A10)
RETURN
END

```

PAUSE

```

SUBROUTINE FWORD (NW,N,LET)
*
*
*.....COSTRUISCE I PUNTORI AI RISPETTIVI GRUPPI DI PAROLE IN
* ORDINE ALFABETICO
*
*
DIMENSION NW(250),LET(250)
INTEGER ALFA
COMMON/ONE/ ALFA(26)
J=0
MASK=77B
DO 100 I=1,N
IL=(NW(I)*8)*8
IL=IL.AND.MASK
IF(I.EQ.1) GO TO 400
IF(IL.EQ.NWT) GO TO 100
400 J=J+1
IF(IL.NE.ALFA(J)) GO TO 300
NWT=IL
LET(J)=1
GO TO 100
300 LET(J)=0
GO TO 400
100 CONTINUE
IF(J.EQ.26) GO TO 500
JJ=J+1
DO 600 I=JJ,26
LET(I)=0
600 CONTINUE
500 RETURN
END

```

```

SUBROUTINE PUNT (K,L,M,N,I)
*
*
*.....COSTRUISCE I PUNTORI DALLA LINGUA INGLESE ALLE ALTRE LINGUE
* E VICEVERSA
*
*
DIMENSION K(250),M(250),L(250)
DO 100 II=1,N
IF(M(II).NE.K(II)) GO TO 100
L(II)=I
GO TO 200
100 CONTINUE
200 RETURN
END

```

```

SUBROUTINE RIGA (N,L1)
*
*
*   INSERISCE UNA RIGA NELLA MATRICE DI CLASSIFICAZIONE
*
*
INTEGER COD
COMMON/UNO/ MAT(250,8)
COMMON/TWO/ COD,IUG
COMMON/SETTE/ ITEMP(8)
COMMON/ING/ EA(250),EB(250),EC(250),ED(250)
WRITE(6,1)
1  FORMAT(///)
   IF(COD.NE.1) GO TO 100
   M=N
400  MM=M-1
     DO 150 J=1,8
150  MAT(M,J)=MAT(MM,J)
     IF(M.EQ.L1) GO TO 300
     M=M-1
     GO TO 400
300  DO 500 J=1,8
500  MAT(M,J)=ITEMP(J)
     GO TO 1000
100  DO 600 I=L1,N
     L2=L1+1
     DO 700 J=1,8
700  MAT(L1,J)=MAT(L2,J)
600  CONTINUE
1000 DO 200 I=1,N
200  CONTINUE
     RETURN
     END

```

```

SUBROUTINE ORDIN(L,N,M,K,NN)
*
*
*   .....ORDINA ALFABETICAMENTE LA LISTA DELLE PAROLE CHIAVE
*
*
DIMENSION L(250),M(250),K(250),NN(250)
CALL ORDER(L,N,M,K,NN)
CALL ORDER(K,N,M,L,NN)
CALL ORDER(M,N,K,L,NN)
RETURN
END

```

```

SUBROUTINE MODPUNT (L,K,I,J1,J2)
*
*
*
*
*
ESEGUE L*AGGIORNAMENTO DEL TRADUTTORE DELLE PAROLE CHIAVE
*
*
DIMENSION L(250),K(250)
COMMON/TWO/ COD,JUG
INTEGER COD
IT=I
IR=I
IF(COD.NE.1) GO TO 1000
100  II=IT-1
    IF(IT.EQ.I2) GO TO 150
    L(IT)=L(II)
    IT=IT-1
150  IF(IT.NE.I2) GO TO 100
    L(I2)=I
    IT=I
200  II=IT-1
    IF(IT.EQ.I1) GO TO 250
    K(IT)=K(II)
    IT=IT-1
250  IF(IT.NE.I1) GO TO 200
    K(I1)=I
    DO 300 J=1,I
    IF(J.EQ.I1)GO TO 400
    IF(K(J).GE.I2) K(J)=K(J)+1
    GO TO 300
400  K(J)=I2
300  CONTINUE
    DO 500 J=1,I
    IF(J.EQ.I2)GO TO 600
    IF(L(J).GE.I1) L(J)=L(J)+1
    GO TO 500
600  L(J)=I1
500  CONTINUE
    I=IR
    GO TO 5001
1000 IF(I2.EQ.(IT+1)) GO TO 2000
    DO 1001 I=I2,IT
    LT=I+1
1001 L(I)=L(LT)
2000 IF(I1.EQ.(IT+1))GO TO 1360
    DO 3000 I=I1,IT
    LT=I+1
3000 K(I)=K(LT)
1360 DO 4000 J=1,IT
4000 IF(K(J).GE.I2) K(J)=K(J)-1
    DO 5000 J=1,IT
5000 IF(L(J).GE.I1) L(J)=L(J)-1
    I=IR
5001 RETURN
END

```

```

SUBROUTINE SNOOPY (L,M,N,K,NN,K1,K2,K3,L1,IQ,IRO)
*
*
* AGGIORNA GLI ACCESSI ALLE LISTE DELLE PAROLE CHIAVE
*
*
DIMENSION L(250),M(250),N(250),K(250)
COMMON/TWO/ COD,IUC
INTEGER COD
IF(COD.NE.1) GOTO 100
LS=IQ
IQ1=IQ
IF(K1.NE.0)GO TO 200
IF(IRO.EQ.26)GO TO 300
IR1=IRO+1
K(IR0)=K2
IF(K(IR0).EQ.0) K(IR0)=K3+1
DO 400 J=IR1,26
IF(K(I).EQ.0) GO TO 400
K(I)=K(I)+1
400 CONTINUE
GO TO 350
300 K(26)=IQ
350 LL=K(IR0)
IF(LL.EQ.IQ)GO TO 1111
LT=L(IQ)
MT=M(IQ)
NT=N(IQ)
390 IN1=IQ-1
L(IQ)=L(IN1)
M(IQ)=M(IN1)
N(IQ)=N(IN1)
IF(IN1.EQ.LL) GO TO 380
IQ=IQ-1
GO TO 390
380 L(LL)=LT
M(LL)=MT
N(LL)=NT
1111 L1=LL
GO TO 1000
200 IF(IRO.EQ.26) GO TO 500
IR1=IRO+1
DO 550 I=IR1,26
IF(K(I).EQ.0) GO TO 550
K(I)=K(I)+1
550 CONTINUE
IF(K2.EQ.0) K2=NN
500 DO 19 I=K1,K2
IF(L(IQ)-L(I)) 20,21,19
21 IF(M(IQ)-M(I)) 20,22,19
22 IF(N(IQ)-N(I)) 20,23,19
19 CONTINUE
LIV=K2+1

```

```
      GO TO 24
20    LIV=I
      GO TO 24
23    LIV=I+1
24    LI=LIV
      IF(LIV.EQ.IQ)GO TO 1000
      LT=L(IQ)
      MT=M(IQ)
      NT=N(IQ)
39    INI=IQ-1
      L(IQ)=L(INI)
      M(IQ)=M(INI)
      N(IQ)=N(INI)
      IF(INI.EQ.LIV) GO TO 38
      IQ=IQ-1
      GO TO 39
38    L(LIV)=LT
      M(LIV)=MT
      N(LIV)=NT
      GO TO 1000
100   IF(LI.EQ.NN) GOTO 1150
      LS=NN-1
      DO 1100 I=L1,LS
      LL=I+1
      L(I)=L(LL)
      M(I)=M(LL)
      N(I)=N(LL)
1100  CONTINUE
1150  IF(IRO.EQ.26) GO TO 1200
      IF(K1.NE.K2) GO TO 1300
      K(IRO)=0
1300  K4=IRO+1
      DO 1400 I=K4,26
      IF(K(I).EQ.0) GO TO 1400
      K(I)=K(I)-1
1400  CONTINUE
      GO TO 1000
1200  IF(K1.EQ.K2) K(26)=0
1000  IQ=IQT
      RETURN
      END
```

```

SUBROUTINE TEST1
*
*
*.....CONTROLLA L*ESATTA SEQUENZA DELLE LINGUE DELLE PAROLE CHIAVE
* DA INTRODURRE
*
*
COMMON/TWO/ COD,IUG
COMMON/TRE/SG(4),LG(4)
INTEGER COD,SG
II=III=0
DO 100 I=1,4
IF(SG(I).EQ.1) II=II+1
IF(SG(I).EQ.0) III=III+1
100 CONTINUE
IF(II.NE.4) GO TO 200
COD=1
GO TO 300
200 IF(III.NE.4) GO TO 400
COD=0
300 IF(LG(1).NE.1) GO TO 600
IF(LG(2).NE.2) GO TO 600
IF(LG(3).NE.3)GOTO 600
IF(LG(4).NE.4) GO TO 600
GOTO 1000
600 WRITE(6,2)
STOP PAUSE
400 WRITE(6,1)
STOP PAUSE
1 FORM1(5X,*VVV INPUT ERROR - CODICE DI AGGIORNAMENTO *
**ERRATO VVV*)
2 FORM1(5X,*VVV INPUT ERROR - ERRORE NELLA SEQUENZA *
**DELLE LINGUE VVV*)
1000 RETURN
END

```

```
SUBROUTINE TEST2 (L,M,N,K,NN,K1,K2,K3,L1,IQ,IRO)
```

```
*
*
*
*
*
```

```
CONTROLLA L*ESISTENZA DELLA PAROLA CHIAVE IN ESAME CON RIFERIMENTO
ALLA LINGUA INGLESE
```

```
DIMENSION L(250),M(250),N(250),K(250)
```

```
COMMON/ONE/ ALFA(26)
```

```
COMMON/DUE/IU
```

```
COMMON/TWO/ COD,IUG
```

```
COMMON/SETTE/ ITEMP(8)
```

```
INTEGER ALFA,COD
```

```
IU=IU+1
```

```
IQ=NN+1
```

```
IUG=IUG+1
```

```
K3=0
```

```
MASK=77H
```

```
IL=(L(IQ)*8)*8
```

```
IL=IL.AND.MASK
```

```
DO 100 I=1,26
```

```
IF(IL.EQ.ALFA(I)) GOTO 200
```

```
IF(K(I).EQ.0) GOTO 100
```

```
K3=K(I)
```

```
100 CONTINUE
```

```
200 K1=K(I)
```

```
IRO=I
```

```
IF(K1.NE.0) GOTO 250
```

```
IF(COD.EQ.0) GOTO 750
```

```
IF(I.NE.26) GOTO 400
```

```
K2=0
```

```
IF(IU.NE.4) GOTO 800
```

```
GOTO 1000
```

```
250 IF(I.NE.26) GOTO 400
```

```
GOTO 370
```

```
400 J=I+1
```

```
350 DO 300 JJ=J,26
```

```
IF(K(JJ).NE.0) GO TO 500
```

```
300 CONTINUE
```

```
370 K2=NN
```

```
GOTO 550
```

```
500 K2=K(JJ)-1
```

```
550 IF(K1.EQ.0) GOTO 1100
```

```
DO 700 IN=K1,K2
```

```
IF(L(IQ).NE.L(IN)) GOTO 700
```

```
IF(M(IQ).NE.M(IN)) GOTO 700
```

```
IF(N(IQ).NE.N(IN)) GOTO 700
```

```
L1=IN
```

```
IF(COD.NE.1) GOTO 800
```

```
IF(IUG.NE.1) GO TO 800
```

```
WRITE(6,5)
```

```
STOP
```

```
700 CONTINUE
```

```
PAUSE
```

```
      IF(IU.NE.4) GO TO 800
      IF(COD.EQ.1) GO TO 1000
750   WRITE(6,1) L(IQ),M(IQ),N(IQ)
      STOP
1100  IF(COD.NE.1) GOTO 800
      IF(IU.NE.4) GOTO 800
1000  READ(5,3) (ITEMP(I),I=1,8)
1     FORMAT (5X,*VVV INPUT ERROR - CHIAVE SCONOSCIUTA VVV*,5X,3A10)
3     FORMAT(7A10,A3)
5     FORMAT(5X,*VVV INPUT ERROR - CHIAVE GIA ESISTENTE NELLA LINGUA*,
**INGLESE VVV*)
800   RETURN
      END
```

PAUSE

```

SUBROUTINE TEST3(EA,EB,EC,K,NN,L1)
*
*
* CONTROLLA NELLA FASE DI AGGIORNAMENTO DI ELEMENTI DELLA MATRICE
* DI CLASSIFICAZIONE E DI RICERCA DELL*UTENTE L*ESISTENZA DELLA
* PAROLA CHIAVE IN ESAME CON RIFERIMENTO ALLA LINGUA INGLESE
*
*
DIMENSION EA(250),EB(250),EC(250),K(250)
INTEGER EEA,EEB,EEC
INTEGER ALFA,COD,EA,EB,EC
COMMON/ONE/ ALFA(26)
COMMON/TWO/ COD,IUG
COMMON/SEI/ LP
COMMON/THREE/EEA,EEB,EEC
IF(COD.NE.7)GO TO 150
READ(5,10)EEA,EEB,EEC,LP
GO TO 160
150 IF(COD.NE.6) GO TO 900
160 MASK=77F
    IL=(EEA*8)*8
    IL=IL.AND.MASK
    DO 100 I=1,26
    IF(IL.EQ.ALFA(I))GO TO 200
100 CONTINUE
200 K1=K(I)
    IF(K1.EQ.0)GO TO 250
    IF(I.NE.26)GO TO 400
    GOTO 370
400 J=I+1
350 DO 300 JJ=J,26
    IF(K(JJ).NE.0)GO TO 500
300 CONTINUE
270 K2=NN
    GO TO 550
500 K2=K(JJ)-1
550 DO 700 IN=K1,K2
    IF(EEA.NE.EA(IN))GO TO 700
    IF(EEB.NE.EB(IN))GO TO 700
    IF(EEC.NE.EC(IN))GO TO 700
    L1=IN
    GO TO 800
700 CONTINUE
250 IF(COD.EQ.6) GOTO 750
    WRITE(6,1) EEA,EEB,EEC
    STOP
750 WRITE(6,2)EEA,EEB,EEC
    STOP
900 WRITE(6,3)
    STOP
1  FORMAT(1X,*VVV INPUT ERROR - CHIAVE NON ESISTENTE VVV*,
*3A10)
2  FORMAT(1X,*VVV INPUT ERROR - AGGIORNAMENTO CLASSIFICAZIONE*,

```

PAUSE

PAUSE

PAUSE

```

** SU CHIAVE NON ESISTENTE VVV*,2A10)
3   FORMAT(1X,*VVV INPUT ERROR - CODICE ERRATO DELLA CHIAVE CUI*,
** AGGIORNARE LA CLASSIFICAZIONE VVV*)
10  FORMAT(3A10,R1)
800 RETURN
    END

```

```

SUBROUTINE AGCLAS (NP,NL,LEJ)

```

```

*
*
*
*
*

```

```

AGGIORNA UN ELEMENTO DELLA MATRICE DI CLASSIFICAZIONE

```

```

COMMON/UNO/ MAT(250,8)
COMMON/TWO/ COD,IUG
INTEGER COD
MASK=77B
IF (NL.NE.0) GO TO 800
MPAR=MAT(LE1,NP)
GO TO 900
800 MPAR=MAT(LE1,NP+1)
    DO 100 I=1,NL
100 MPAR=(MPAR*8)*8
900 MPART=MPAR
    MPART=MPART.AND.MASK
    IF (COD.EQ.0) COD=33B
    IF (COD.EQ.1) COD=34B
    IF (MPART.EQ.COD) GOTO 500
    IF (MPART.GT.COD) GOTO 300
    MPAR=MPAR.OR.34B
    MPAR=MPAR.AND.(.NOT.03B)
    GO TO 400
300 MPAR=MPAR.OR.33B
    MPAR=MPAR.AND.(.NOT.44B)
400 IF (NL.EQ.0) GO TO 450
    NL=10-NL
    DO 150 II=1,NL
150 MPAR=(MPAR*8)*8
    MAT(LE1,NP+1)=MPAR
    GO TO 500
450 MAT(LE1,NP)=MPAR
500 RETURN
    END

```

D A T I D I I N P U T

CODICE DELLA LINGUA J

PAROLE CHIAVE

MATRICE
 COMPILATORE
 PLESSI
 CODEEPIL
 LINGUAGGI DI PROGRAMMAZIONE

D A T I D I O U T P U T

CLASSE INDIVIDUATA

22 LINGUAGGI DI PROGRAMMAZIONE NUMERO PER STAMPA LISTA LIBRI 9

D A T I D I I N P U T

CLASSE DI LIBRI RICHIESTA 9

D A T I D I O U T P U T

LISTA DEI LIBRI

22 LINGUAGGI DI PROGRAMMAZIONE		
R -22 -4 LAUTIER D.-LERNER J.P.	MASSON	73
COURS DE BASIC - ANALYSE ET PROGRAMMATION		
R -22 -2 MAURER W.D.	MACDONALD	72
THE PROGRAMMER'S INTRODUCTION TO LISP		
R -22 -2 RALSTON A.	MCGRAW-HILL	71
FORTRAN IV PROGRAMMING. A CONCISE EXPOSITION		
-22 -15 AMBROSIO S.	BORINGHIERI	65
LINGUAGGI ALGEBRICI		
-22 -14 BOLT A.B.-WARDLE M.E.	U.CAMBRIGE	70
COMMUNICATING WITH A COMPUTER		
-22 -12 ERSHOV A.P.	PERGAMON PRESS	59
PROGRAMMING PROGRAMME FOR THE BESM COMPUTER		
-22 -5 GOODMAN	PERGAMON PRESS	60
ANNUAL REVIEW AUTOMATIC PROGRAMMING		
-22 -8 LAFITTE R.	DUNOD	69
INITIATION AU LANGAGE SCHOOL		
-22 -7 MC KRACKEN D.	WILEY J.-SONS	61
A GUIDE TO FORTRAN PROGRAMMING		
-22 -13 NOLIN L.	GAUTHIER-VILLARS	69
FORMALISATION DES NOTIONS DE MACHINE ET DE PROGRAMME		
-22 -16 PETERSON	WILEY J.-SONS	61
ERROR - CORRECTING CODES		
-22 -11 REVIGLIO G.	BORINGHIERI	64
I LINGUAGGI DEGLI ELABORATORI ELETTRONICI		
-22 -9 VALSESIA S.	BORINGHIERI	65
EVOLUZIONE DEI LINGUAGGI DEGLI ELABORATORI ELETTRONICI		

D A T I D I I N P U T

CODICE DELLA LINGUA E

PAROLE CHIAVE

RETADISTRIBUTION
 BINOMIALRV
 CUMULATIVEDISTRIBUTIONFUNCTION
 GAMMADISTRIBUTION
 HYPERGEOMETRICRV

D A T I D I O U T P U T

CLASSI INDIVIDUATE

VVV CHIAVI NON SELETTIVE VVV

11 CALCOLO DELLE PROBABILITA' NUMERO PER STAMPA LISTA LIBRI 12
 22 DISTRIBUZIONI CAMPIONARIE NUMERO PER STAMPA LISTA LIBRI 13

D A T I D I I N P U T

CLASSE DI LIBRI RICHIESTA 13

D A T I D I O U T P U T

LISTA DEI LIBRI

22 DISTRIBUZIONI CAMPIONARIE
 R -32 -1 AMATO V. GIUFFRE' DOTT.A. 58
 FORME QUADRATICHE IN STATISTICA
 RA -32 -6 ELDERTON W.P.-JOHNSON N.L. V.CAMBRIDGE PRESS 69
 SYSTEMS OF FREQUENCY CURVES
 R -32 -3 KENDALL M.G. GRIFFIN 72
 A COURSE IN MULTIVARIATE ANALYSIS, REJNG NUMBER TWO
 R -32 -5 MARDIA K.V. GRIFFIN 70
 FAMILIES OF BIVARIATE DISTRIBUTIONS, REJNG NUMBER TWENTY-SEVEN
 R -32 -2 NEWMAN T.G.-ODELL P.L. GRIFFIN 71
 THE GENERATION OF RANDOM VARIATES, REJNG NUMBER TWENTY-NINE
 R -32 -4 ORD J.K. GRIFFIN 72
 FAMILIES OF FREQUENCY DISTRIBUTIONS, REJNG NUMBER THIRTY
 RA -32 -7 STICHPROBEN STAT.BUNDESAMT 60
 ...IN DER AMTLICHEN STATISTIK
 -32 -8 AITCHISON J.-BROWN J.A.C. V.CAMBRIDGE PRESS 57
 THE LOGNORMAL DISTRIBUTION
 -32 -9 COCHRAN W. WILEY J. - SONS 53
 SAMPLING TECHNIQUES

5. - CONSIDERAZIONI E PROSPETTIVE PER NUOVE RICERCHE.

5.1 *Considerazioni sulla metodologia usata.*

La vastità dei problemi che si sono dovuti affrontare per realizzare l'interprete del linguaggio LINGEB ha imposto di compiere delle scelte precise nella costruzione degli algoritmi da impiegare.

L'obiettivo iniziale era quello di costruire degli algoritmi tali da realizzare contemporaneamente una fase di costruzione rapida ed altrettanto rapide fasi di aggiornamento e ricerca.

Nella realizzazione pratica si è invece constatato l'impossibilità di ottenere quanto sopra, essendo la fase di costruzione in antitesi alle altre.

Si è, pertanto, preferito costruire degli algoritmi atti ad ottimizzare le fasi di aggiornamento e di ricerca.

Questa scelta è motivata dal fatto che si sono ritenute essere queste le fasi che vengono usualmente impiegate nella gestione automatica di una biblioteca, poiché le fasi di costruzione si rendono necessarie solo inizialmente per organizzare i dati di base.

Si è dovuto compiere un'altra scelta nella fase di aggiornamento dell'archivio dei libri al riguardo dell'eventuale cancellazione di dati relativi a libri non più esistenti nella biblioteca per smarrimento o per altre cause.

Si è preferito non procedere alla loro cancellazione, ma considerarli invece come ancora esistenti.

In questo modo ci si è uniformati a quelli che sono i criteri usati nelle gestioni tradizionali di biblioteche.

Infine le fasi relative all'utente sono state organizzate in modo che egli possa recuperare i dati relativi a tutti i libri che appartengono alla classe individuata tramite la ricerca da lui effettuata.

Ciò è giustificato dal fatto che normalmente una biblioteca è utilizzata da persone che desiderano avere la possibilità di essere informati su quello che esiste di un determinato argomento, ed eventualmente scegliere personalmente l'indirizzo specifico da dare alla propria ricerca.

Per quanto riguarda il « thesaurus » si è adottata una realizzazione basata su una metodologia semplice, onde poterlo costruire nel tempo che si rendeva disponibile, poiché per ottenere un « thesaurus » più sofisticato e più efficiente sarebbero stati necessari degli approfonditi studi linguistici.

5.2 *Valutazioni sul sistema realizzato.*

Un sistema di gestione automatica di una biblioteca può essere valutato in base alle seguenti caratteristiche:

- 1) qualità e quantità delle informazioni memorizzate in relazione allo specifico campo di applicazione;
- 2) capacità del sistema di recuperare le informazioni richieste (potere di richiamo);
- 3) capacità del sistema di evitare il ritrovamento di informazioni non pertinenti (potere di precisione);
- 4) tempo medio di risposta;
- 5) facilità d'utilizzo;
- 6) costo di realizzazione e gestione.

Con riferimento al sistema descritto nei precedenti capitoli si fa osservare che per quanto concerne il punto 1), mentre per il « thesaurus » si è già parlato nel paragrafo precedente, relativamente alle informazioni sui libri si è deciso di memorizzare, almeno per ora, una parte di quelle che vengono normalmente considerate utili per la loro individuazione.

Per i punti 2), 3) e 4) non si è in grado di compiere le necessarie valutazioni, poiché esse possono essere fatte solo dopo la memorizzazione dei dati di base relativi a tutte le classi che compongono la biblioteca, mentre, per ora, ciò è stato realizzato solo per una parte di esse.

Per il punto 5) si ritiene che il sistema realizzato sia facilmente utilizzabile dall'utente, avendo quest'ultimo a disposizione un insieme di comandi molto semplice.

Infine per quanto riguarda il punto 6) non sono state effettuate valutazioni di costi, non essendo questo, almeno per ora, lo scopo primario della ricerca.

Comunque si può fin d'ora osservare che non è possibile ottimizzare un sistema rispetto a tutti i punti elencati, ma che è invece necessario studiare volta per volta quali sono quelli più interessanti con riferimento al particolare problema che si sta risolvendo.

5.3 *Possibili sviluppi della ricerca.*

Nel sistema che è stato realizzato, i dati relativi ai vari libri sono stati suddivisi conformemente alla classificazione già esistente nella biblioteca oggetto di studio.

Qualsiasi nuovo arrivo necessita quindi di una preventiva analisi da parte di un esperto che ha il compito di decidere la classe in cui collocarlo. Questo comporta una serie di inconvenienti dovuti ad una scelta necessariamente soggettiva.

Per ovviare, almeno in parte, a queste difficoltà si trovano in letteratura dei metodi di classificazione « a posteriori » nei quali il parere dell'esperto non è determinante, ma subisce una successiva verifica automatica consistente nel controllare se determinate parole scelte nel contesto del libro individuano effettivamente la classe da lui indicata.

Questo metodo, tuttavia, presenta il limite notevole che anche in sede di reperimento delle parole dal contesto necessita nuovamente dell'intervento di esperti e quindi rimane ancora legato a giudizi soggettivi.

Un approccio diverso che può permettere di ottenere la classificazione di un libro in modo autonomo dal giudizio soggettivo degli esperti è quello di procedere ad una classificazione a priori.

Questo metodo consiste nell'estrarre dal libro un certo numero di informazioni e di classificarlo in base ad esse.

Il titolo, per esempio, può fornire utili indicazioni per la classificazione del libro.

Tuttavia questa informazione molto spesso è insufficiente, per cui si può fare ricorso ad altre fonti, quali ad esempio l'indice, la bibliografia o un breve riassunto del testo.

Al riguardo non si ha, per ora, notizia di alcuna realizzazione pratica, per cui può costituire un interessante campo di futura ricerca.

BIBLIOGRAFIA

- [1] A.I.C.A. - *Compilatori e tecniche di implementazione*. Editrice Tecnico-Scientifica, Pisa, 1970.
- [2] A. ANDRONICO - *Scienza degli elaboratori*. Vol. 1°. Ed. Zanichelli, Bologna, 1973.
- [3] A. CHANDOR - *Dizionario di informatica*. Ed. Zanichelli, Bologna, 1972.
- [4] E. CRESCENTI, L. COLUSSI, F. GOSEN - *Teoria e tecnica della elaborazione automatica dei dati*. Ed. CLEUP, Padova, 1974.
- [5] F. DALLA LIBERA - *Analisi sintattica e compilatori*. Tesi di laurea. Università degli studi di Padova, A.A. 1969-70.
- [6] D. DA RIVA, G. RICCI - *Elementi di organizzazione delle informazioni*. Ed. Franco Angeli, Milano, 1972.
- [7] B. DOUDNIKOFF - *Information Retrieval*. Ed. Auerbach, Philadelphia, 1973.
- [8] G. FIORENZA - *Costruzione e riconoscimento di grammatiche e di parole di un linguaggio mediante APL*. Tesi di laurea. Università degli Studi di Padova, A. A. 1968-1969.
- [9] D. GRIES - *Compiler construction for digital computers*. Ed. John Wiley, New York, 1971.
- [10] A. HERRMAN, G. VALLE - *Un programma per la gestione di una base di dati con tecniche relazionali*. C.N.R., Centro di studio per l'interazione operatore-calcolatore, Bologna, 1974.
- [11] J. E. HOPCROFT, J. D. ULLMON - *Formal languages and their relation to automata*. Ed. Addison-Wesley, New York, 1969.
- [12] F. R. A. HOPGOOD - *Compiling Techniques*. Ed. American Elsevier Inc., New York, 1969.
- [13] K. S. JONES - *Automatic Keyword classification for information retrieval*. Ed. Butterworths, Londra, 1971.
- [14] K. S. JONES - *Automatic term classification and information retrieval*. In: « Information processing », 68, pagg. 1290-1295. Ed. North-Holland Co., Amsterdam, 1969.
- [15] D. E. KNUTH - *The art of computer programming*. Vol. 1, 3. Ed. Addison-Wesley, New York, 1968.
- [16] K. KOYMEN - *A text organizing system*. Vol. 1, 2. University of Pennsylvania, Philadelphia, 1974.
- [17] A. L. LANG, S. W. ZAGORSKY - *Implementation of an automatic, a posteriori, hierarchical classification system*. University of Pennsylvania, Philadelphia, 1972.

- [18] B. LANGEFORS - *Theoretical analysis of information systems*. Ed. Auerbach, Philadelphia, 1973.
- [19] I. C. LERMAN - *Les bases de la classification automatique*. Ed. Gauthier-Villars, Parigi, 1970.
- [20] B. LITOFESKY - *Utility of automatic classification systems for information storage and retrieval*. University of Pennsylvania, Philadelphia, 1969.
- [21] F. LUCCIO - *Strutture linguaggi e sintassi*. Ed. Boringhieri, Torino, 1974.
- [22] N. S. PRYWES - *All-automatic processing for a large library*. In: « American Federation of Information Processing Societies », pagg. 323-333. Ed. AFIPS, Atlantic City, 1970.
- [23] N. S. PRYWES, A. L. LANG, S. ZAGORSKY - *A posteriori indexing, classification and retrieval of textual data*. In: « Information Storage and Retrieval », vol. 10, pagg. 15-17. Ed. Pergamon Press, Londra, 1974.
- [24] M. A. ROTHMAN - *La rivoluzione cibernetica*. Ed. Rizzoli, Milano, 1973.
- [25] G. SALTON - *The SMART retrieval system. Experiments in automatic document processing*. Ed. Prentice-Hall, Englewood Cliffs, 1971.
- [26] Scienza e tecnica « 70 - Annuario della EST ». *Natura e sviluppo della informazione scientifica*. Ed. Mondadori, Milano, 1970.
- [27] G. TONINELLI, G. VALLE - *Tecniche relazionali per la gestione dei dati*. C.N.R., Centro di studio per l'interazione operatore-calcolatore, Bologna, 1973.
- [28] P. L. WHITE - *Kwic/360-Keyword in context indexing program for the IBM system/360*. IBM (U.K.), Wilmslow, 1960.
- [29] M. WOLFF-TERROINE, D. RIMBERT, B. RONAUULT - *La classification automatique, son utilisation pour la constitution du langage et l'interrogation des systems documentaires*. In: « Automatique informatique recherche operationnelle ». Ed. Dunod, Parigi, 1972.

