

ISTITUTO DI ELABORAZIONE DELLA INFORMAZIONE - C.N.R. - PISA

ALCUNE NOTE SUL LINGUAGGIO SEL,

(versione 8K),

E SULLA SUA IMPLEMENTAZIONE.

G. Molnar - R. Sprugnoli

A. Carobbi - E. Ricciardi

Nota Interna n° B-70-6

Premessa

Si ricordano le principali caratteristiche del SEL.

Il SEL é un linguaggio autoestensibile nel senso che un programma scritto in SEL può ampliare le capacità del linguaggio stesso.

Un programma SEL é un insieme ordinato di oggetti. Un oggetto si compone di un oggetto esterno e di un oggetto interno.

L'oggetto esterno é una parola che identifica l'oggetto stesso, quindi ha un nome che può avere diverse occorrenze nel testo del programma. L'oggetto interno é formato da una procedura (se non altrimenti definita essa consiste di un messaggio d'errore) che viene eseguita nel caso di un accesso diretto all'oggetto, e da un valore di tipo opzionale.

Il SEL possiede un insieme di oggetti base le cui procedure rappresentano le istruzioni fondamentali del SEL.

In ogni istante il sistema SEL si trova in uno di due stati: interpretativo o generativo. Per quanto riguarda le istruzioni fondamentali, se lo stato é interpretativo esse vengono eseguite quando si abbia un accesso diretto, mentre se lo stato é generativo viene generata una procedura che rappresenta l'istruzione stessa.

Un insieme ordinato di procedure generate (una macro) é un oggetto interno, quindi può essere assegnato ad un oggetto esterno (nome). Le macro possono avere dei "parametri formali" per mezzo di un meccanismo di riferimento che permette di calcolare i loro valori sia durante la generazione che durante l'esecuzione della macro stessa. Un particolare riferimento é quello all'oggetto il cui nome é successivo a quello al quale appartiene la macro come oggetto interno.

Per una descrizione più dettagliata vedi [1] .

Organizzazione generale

L'esecuzione di un programma SEL consiste nell'esecuzione delle procedure dei singoli oggetti che formano il programma stesso. Per far ciò sono necessarie due cose:

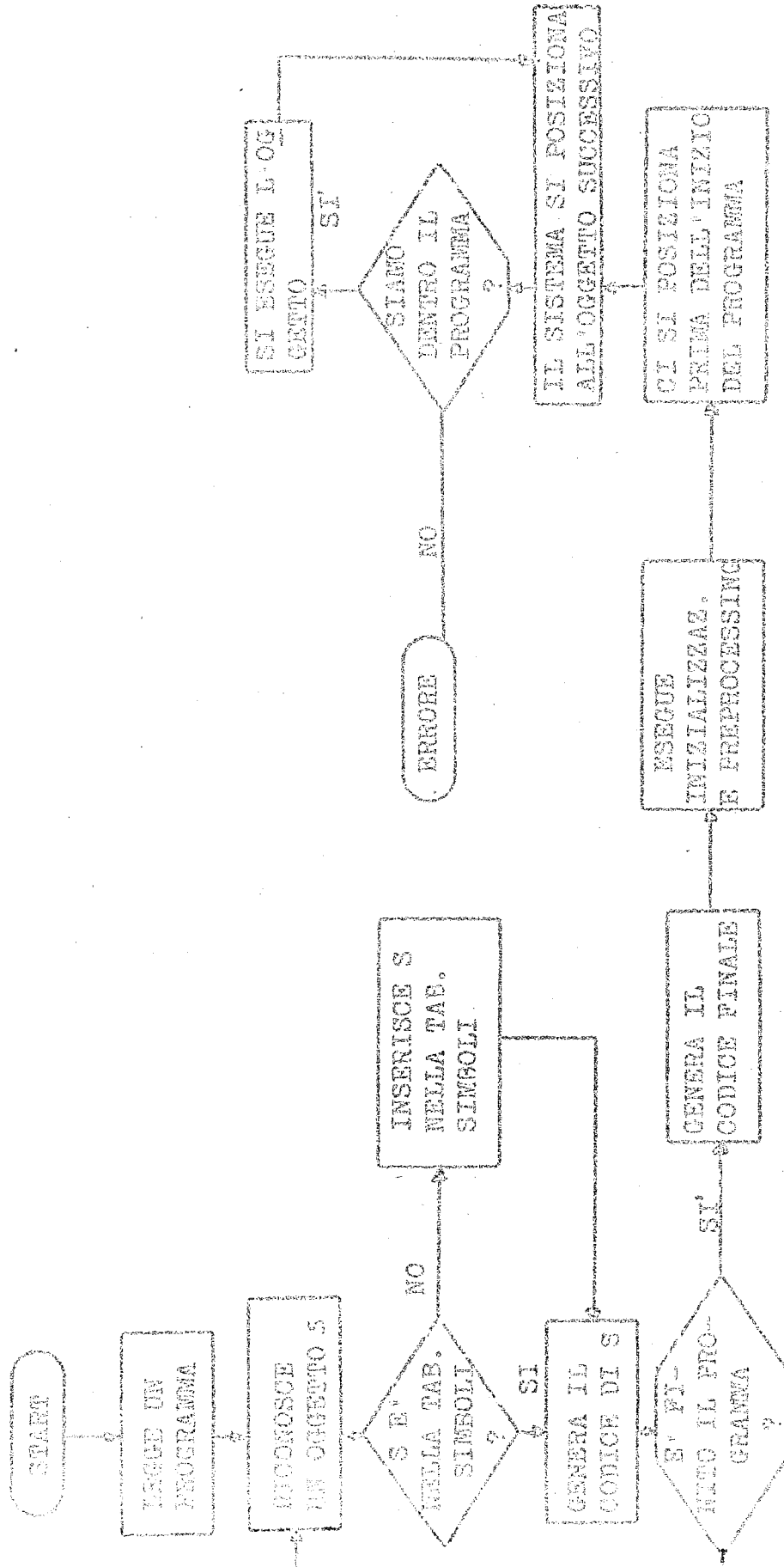
- a) la capacità da parte del sistema di introdurre nel sistema stesso nuovi oggetti esterni;
- b) l'esistenza di un supervisore che una volta eseguite la procedura di un oggetto passi all'esecuzione della procedura dell'oggetto logicamente successivo.

Ciò è stato realizzato mediante un programma il cui diagramma a blocchi è dato in figura; in tale diagramma i due punti a) e b) corrispondono ai lati sinistro e destro, rispettivamente, della figura.

Il programma corrispondente al punto a) fa uso di un "buffer di lettura" e genera due tabelle, dette rispettivamente "tabella dei simboli" e "tabella dei codici"; il programma corrispondente al punto b) usa le tabelle generate dal programma precedente, nonché due zone di memoria, dette "zona dei programmi generati" e "zona dei vettori".

Oltre a queste il programma fa uso di altre tabelle, per i riferimenti e per il sistema conversazionale che vedremo a suo tempo; ne risulta in definitiva una organizzazione di memorie a zone e si è cercato quindi di sfruttare la divisione in pagine della memoria dell'HP-2116B. Come esempio diamo l'organizzazione della memoria nella versione SEL a 8k. (Il buffer di lettura è situato dentro la zona "programmi SEL" a 16000₈, nella figura "organizzazione della memoria ecc.").

DIAGRAMMA A BLOCCHI DELLA ORGANIZZAZIONE DEL SISTEMA "SEI"



Disco una breve descrizione delle tabelle e delle zone di memoria usate dal SEL.

a. Buffer di lettura.

Si tratta di 36 celle di memoria, situate nella stessa pagina (la 7^a) che contiene il programma di lettura ed il programma principale del SEL, e ciò per comodità di implementazione. Tale buffer viene a contenere i caratteri di una riga di programma, due per cella, in quanto per comodità di lettura una riga battuta sulla telescrivente può contenere al massimo 72 caratteri.

b. Tabella dei simboli.

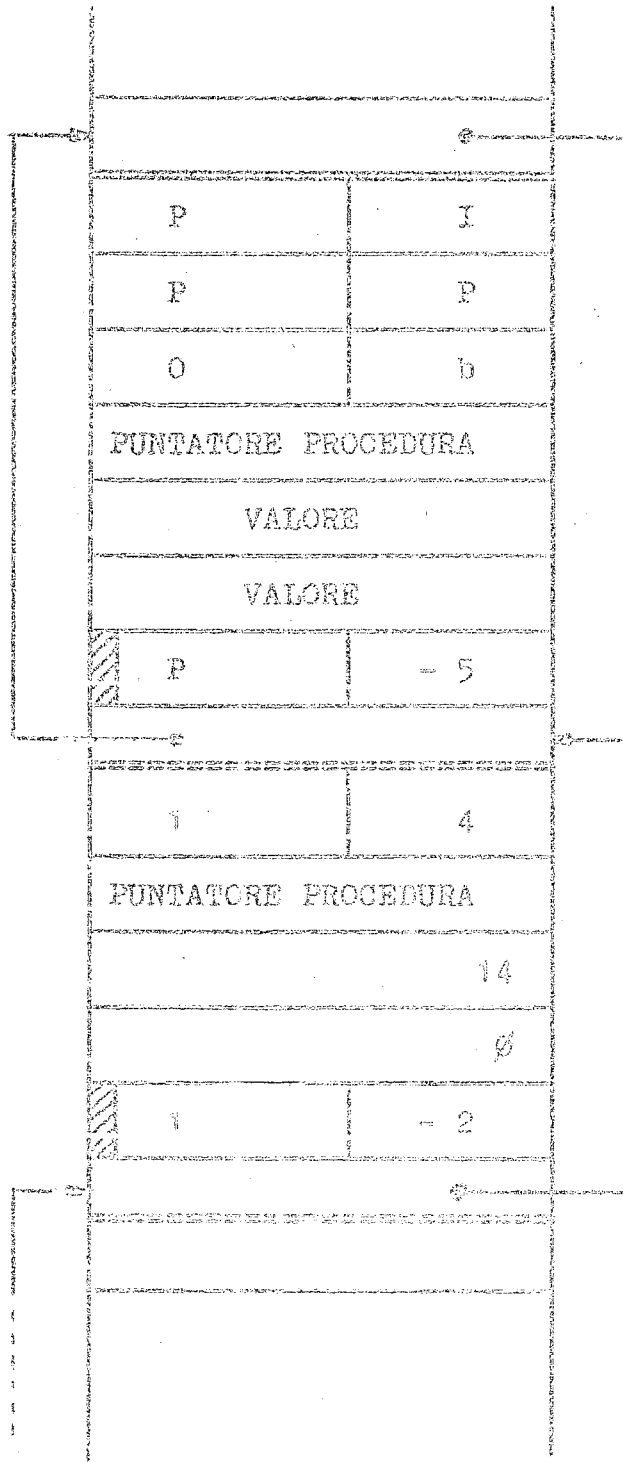
L'organizzazione della tabella dei simboli è schematizzata nella figura relativa. Ad ogni oggetto è assegnato un numero variabile di celle di memoria per contenere tutte le informazioni ad esso pertinenti. Se n è il numero di caratteri che formano il nome dell'oggetto, il numero delle celle di memoria ad esso assegnato è esattamente: $\lceil (n+1)/2 \rceil + 5$, dove con le parentesi quadre si è indicato il massimo intero minore o uguale al numero scritto entro le parentesi stesse.

Partendo dalla parte inferiore della memoria si ha: $\lceil (n+1)/2 \rceil$ celle contengono il nome dell'oggetto, due caratteri per cella; la cella successiva contiene un puntatore all'inizio della procedura assegnata all'oggetto (oggetto interno); se all'oggetto non è assegnata alcuna procedura, in questa cella si trova un puntatore ad una routine d'errore.

Le due celle successive contengono il valore dell'oggetto; in casi speciali possono contenere anche altre informazioni.

Le ultime due celle non si riferiscono propriamente all'oggetto a cui appartengono, ma sono di utilità per i programmi che

ORGANIZZAZIONE DELLA TABELLA DEI SIMBOLI



sono la tabella dei simboli e servono per l'organizzazione della tabella stessa. La prima di queste due celle è divisa in tre parti; il bit del segno non è significativo; i bit 1 - 7 contengono la prima lettera del nome dell'oggetto; i bit 8 - 12 contengono la lunghezza negativa del nome dell'oggetto; il contenuto di questa cella è detto il "peso" del simbolo. Questa cella serve soprattutto per la ricerca di un oggetto nella tabella stessa. La seconda di queste celle, ultima di quelle che appartengono all'oggetto, contiene un puntatore alla cella enciclopedia dell'oggetto successivo; ciò permette di elaborare un metodo semplice per la ricerca sistematica dei nomi degli oggetti.

c. Tabella dei codici.

Man mano che il programma di lettura legge il programma SRL che si vuole eseguire, viene posto in celle successive di questa tabella l'indirizzo nella tabella dei simboli della cella contenente il riferimento alla procedura dell'oggetto esterno che è stato trovato. In tal modo, una volta letto dal supporto esterno un programma SRL, la tabella dei codici viene a contenere il programma stesso in forma compattata; ogni cella rappresenta una occorrenza di un oggetto nel programma. L'esecuzione del programma stesso consisterà quindi nella successiva esecuzione delle procedure degli oggetti il cui riferimento si trova nella tabella dei codici.

d. Area dei vettori.

Questa zona della memoria è inizialmente del tutto libera; ogni definizione di vettore in un programma SRL occupa ad esso vettore una parte di questa zona secondo le dimensioni del vettore stesso; le operazioni su detto vettore provvederanno al riempimento delle

celle ad esso riservate.

e. Zona dei sottoprogrammi generati.

E' questa la zona di memoria destinata a contenere i programmi generati dal SEL. Mentre gli oggetti base del SEL hanno le loro procedure in parti ben determinate della memoria, gli altri oggetti non hanno a priori alcuna procedura che sia loro assegnata, se non una routine di errore. Il SEL prevede due metodi per definire e assegnare procedure agli oggetti (v. "Alcune caratteristiche del SEL"): questa zona di memoria contiene allora le procedure definite con uno di tali metodi in un programma (o più programmi) SEL; si noti però che la definizione di una procedura è del tutto indipendente dalla sua assegnazione a un oggetto esterno, anzi possono esistere delle procedure che non siano assegnate ad alcun oggetto esterno.

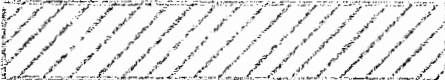
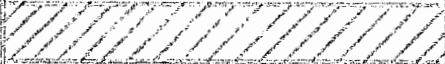
Per ciò che riguarda l'organizzazione della memoria, si è cercato di sfruttare al massimo la struttura a pagine dell'HP2115-B. In figura è riportata l'organizzazione nella versione 8k.

Le pagine 0, 1 e 7 sono state riservate ai programmi del linguaggio SEL. Il programma principale è a pagina 7, col driver che comanda l'input-output del sistema.

Una buona parte della pagina 0 è riservata alle "tabelle di riferimento" cioè ai links fra i vari programmi; questa collocazione è dovuta al particolare ruolo che la pagina 0 ha nel calcolatore.

La pagina 2 è stata riservata alla versione conversazionale del SEL, e di essa parleremo a suo tempo.

ORGANIZZAZIONE DELLA MEMORIA NEL PROGRAMMA SEL - VERSIONE 82

0		PAGINA 0
100	PROGRAMMI SEL	
177	TABELLE DI RIFERIMENTO	
500		
1000	PROGRAMMI SEL	PAGINA 1
4000	TESTO	PAGINA 2
6000	TABELLA DEI CODICI	PAGINA 3
7000	ZONA DEI VETTORI	
10000	ZONA DEI PROGRAMMI GENERATI	PAGINA 4
12000	DUMP	PAGINA 5
12300	TABELLA DEI SIMBOLI	PAGINA 6
16000	PROGRAMMI SEL	PAGINA 7
	DRIVER	
17700	BINARY LOADER	

La pagina 3 é stata suddivisa fra la tabella dei codici e la zona dei vettori. Tale suddivisione permette al SEL di accettare programmi contenenti fino a 512 occorrenze di oggetti, il che permette di elaborare programmi sufficientemente vasti; le 512 celle riservate alla zona dei vettori, possono invece portare alcune limitazioni ai programmi che ne facciano uso.

Tutta la pagina 4 é stata riservata ai programmi generati, con la convenzione che i programmi generati veri e propri vadano ad occupare la pagina da 10000_8 in avanti, mentre le celle riservate ai links necessari agli stessi programmi, invadano la memoria partendo da 11777_8 e procedendo all'indietro.

Le prime 300_8 celle della pagina 5 contengono un programma di servizio, il dump, da usare soprattutto per il debugging dei programmi.

Ciò che resta della pagina 5 e tutta la pagina 6 sono riservate alla tabella dei simboli, per un totale di 1056_{10} celle di memoria. Gli oggetti base del sistema occupano già circa 540_{10} di queste celle, quindi rimangono all'utente circa 131_{10} posizioni; tenuto conto che ogni oggetto viene ad occupare in media circa 8 - 9 celle (nomi di 5 - 8 caratteri) si può dire che tale spazio é sufficiente per circa 15_{10} oggetti.

Si noti che il SEL prevede un sistema di controllo sull'uso di queste zone della memoria affinché non avvengano sovrapposizioni che potrebbero mandare all'aria il sistema stesso.

Il Riconoscitore

Gli oggetti del SEL, come abbiamo visto, sono identificati da un nome; tale nome provvede a identificare i vari tipi di oggetti che il SEL può trattare.

Gli oggetti del SEL si dividono in "costanti" e "variabili"; le costanti sono state a loro volta divise in "interi", "reali" e "stringhe", mentre le variabili sono state divise in "caratteri", "identificatori" e "identificatori speciali".

Per comodità di implementazione, i "reali" sono stati a loro volta divisi in "reali senza esponente" e "reali con esponente". Inoltre, in un programma SEL possono essere introdotti dei commenti, che vengono ignorati dal programma compilatore-esecutore del SEL stesso.

Diamo qui di seguito una definizione formale in notazione di Backus dei nomi che identificano gli oggetti del SEL.

< cifra > ::= \emptyset | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

< lettere meno E > ::= 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 .

< lettera E > ::= E

< separatore > ::= b | $\text{\textcircled{CR}}$

< punto > ::= .

< virgoletto > ::= "

< dollaro > ::= \$

< altro carattere > ::= + | - | * | / | = | > | < | : | ; | , | ! | ? | (|) |
[|] | # | % | & | ' | @ | ^ | \

< delimitatore di commento > ::= \

$\langle \text{plus minus} \rangle ::= + \mid -$
 $\langle \text{lettera} \rangle ::= \langle \text{lettera meno E} \rangle \mid \langle \text{lettera E} \rangle$
 $\langle \text{carattere composto} \rangle ::= := \mid - \mid < : \mid : > \mid < * \mid * >$
 $\langle \text{alfanumero} \rangle ::= \langle \text{cifra} \rangle \mid \langle \text{lettera} \rangle \mid \langle \text{alfanumero} \rangle \langle \text{alfanumero} \rangle$
 $\langle \text{elemento} \rangle ::= \langle \text{cifra} \rangle \mid \langle \text{lettera} \rangle \mid \langle \text{separatoro} \rangle \mid \langle \text{punto} \rangle \mid \langle \text{dollaro} \rangle \mid$
 $\quad \langle \text{altro carattere} \rangle$
 $\langle \text{sequenza} \rangle ::= \langle \text{elemento} \rangle \mid \langle \text{delimitatore di commento} \rangle \mid \langle \text{sequenza} \rangle$
 $\quad \langle \text{sequenza} \rangle$
 $\langle \text{testo di commento} \rangle ::= \langle \text{elemento} \rangle \mid \langle \text{virgolette} \rangle \mid \langle \text{testo di commento} \rangle$
 $\quad \langle \text{testo di commento} \rangle$
 $\langle \text{commento} \rangle ::= \langle \text{delimitatore di commento} \rangle \langle \text{testo di commento} \rangle$
 $\quad \langle \text{delimitatore di commento} \rangle$
 $\langle \text{carattere} \rangle ::= \langle \text{altro carattere} \rangle \mid \langle \text{carattere composto} \rangle$
 $\langle \text{identificatore} \rangle ::= \langle \text{lettera} \rangle \mid \langle \text{lettera} \rangle \langle \text{alfanumero} \rangle$
 $\langle \text{intero} \rangle ::= \langle \text{cifra} \rangle \mid \langle \text{cifra} \rangle \langle \text{intero} \rangle$
 $\langle \text{reale senza esponente} \rangle ::= \langle \text{intero} \rangle \langle \text{punto} \rangle \mid \langle \text{intero} \rangle \langle \text{punto} \rangle \langle \text{intero} \rangle$
 $\langle \text{stringa} \rangle ::= \langle \text{virgolette} \rangle \langle \text{sequenza} \rangle \langle \text{virgolette} \rangle$
 $\langle \text{identificatore speciale} \rangle ::= \langle \text{dollaro} \rangle \mid \langle \text{dollaro} \rangle \langle \text{alfanumero} \rangle$
 $\langle \text{reale con esponente} \rangle ::= \langle \text{reale senza esponente} \rangle \langle \text{lettera E} \rangle$
 $\quad \langle \text{plusminus} \rangle \langle \text{intero} \rangle$
 $\langle \text{costante} \rangle ::= \langle \text{intero} \rangle \mid \langle \text{reale senza esponente} \rangle \mid \langle \text{reale con esponente} \rangle \mid$
 $\quad \langle \text{stringa} \rangle$
 $\langle \text{variabile} \rangle ::= \langle \text{carattere} \rangle \mid \langle \text{identificatore} \rangle \mid \langle \text{identificatore speciale} \rangle$
 $\langle \text{nome} \rangle ::= \langle \text{costante} \rangle \mid \langle \text{variabile} \rangle$

Il SEL é stato provvisto di un riconoscitore di questo linguaggio, riconoscitore che ora vogliamo brevemente descrivere.

Le definizioni < cifra > ... < delimitatore di commento > determinano una divisione in 9 classi dell'insieme dei caratteri a disposizione dell'utente della telescrivente, apparecchiatura d'ingresso del sistema SEL.

Alle classi < commento > ... < reale con esponente > è stato assegnato un "numero di stato" variabile da 0 a 7, nell'ordine dato dalle definizioni.

Sono state infine definite le seguenti 6 "azioni" che il riconoscitore deve intraprendere in funzione dello stato in cui si trova e del carattere che sta leggendo:

- CN : continua; aggiungi il carattere letto al nome che stai formando e leggi un altro carattere;
- IG : ignora; ignora il carattere letto e leggi un altro;
- TC : termina e comprendi; aggiungi il carattere letto al nome che stai formando e considera completato tale nome;
- TI : termina e ignora; ignora il carattere che hai letto e considera completato il nome che stai formando;
- TE : termina e escludi; considera completato il nome che stai formando e prendi il carattere letto come appartenente all'oggetto successivo;
- ES : esponente; si è incontrata la lettera E dopo un reale senza esponente; se è seguita da "+" o "-" inizia un esponente, altrimenti la E è l'inizio di un identificatore.

Il riconoscitore è allora costituito da un programma semplicissimo. Si parte da uno stato convenzionale (1), si legge un carattere e quindi si cambia stato o si esegue una determinata azione secondo la seguente tabella:

	∅	1	2	3	4	5	6	7
< cifra >	IG, ∅	CN, 3	CN, 2	CN, 3	CN, 4	CN, 5	CN, 6	CN, 7
< lettera meno E >	IG, ∅	CN, 2	CN, 2	TE, 3	TE, 4	CN, 5	CN, 6	TE, 7
< lettera E >	IG, ∅	CN, 2	CN, 2	TE, 3	ES, 4	CN, 5	CN, 6	TE, 7
< separatore >	IG, ∅	IG, 1	TI, 2	TI, 3	TI, 4	CN, 5	TI, 6	TI, 7
< punto >	IG, ∅	TC, 1	TE, 2	CN, 4	TE, 4	CN, 5	TE, 6	TE, 7
< virgolette >	IG, ∅	IG, 5	TE, 2	TE, 3	TE, 4	TI, 5	TE, 6	TE, 7
< dollaro >	IG, ∅	CN, 6	TE, 2	TE, 3	TE, 4	CN, 5	TE, 6	TE, 7
< altro carattere >	IG, ∅	TC, 1	TE, 2	TE, 3	TE, 4	CN, 5	TE, 6	TE, 7
< delim. di com. >	IG, 1	IG, ∅	TE, 2	TE, 3	TE, 4	CN, 5	TE, 6	TE, 7

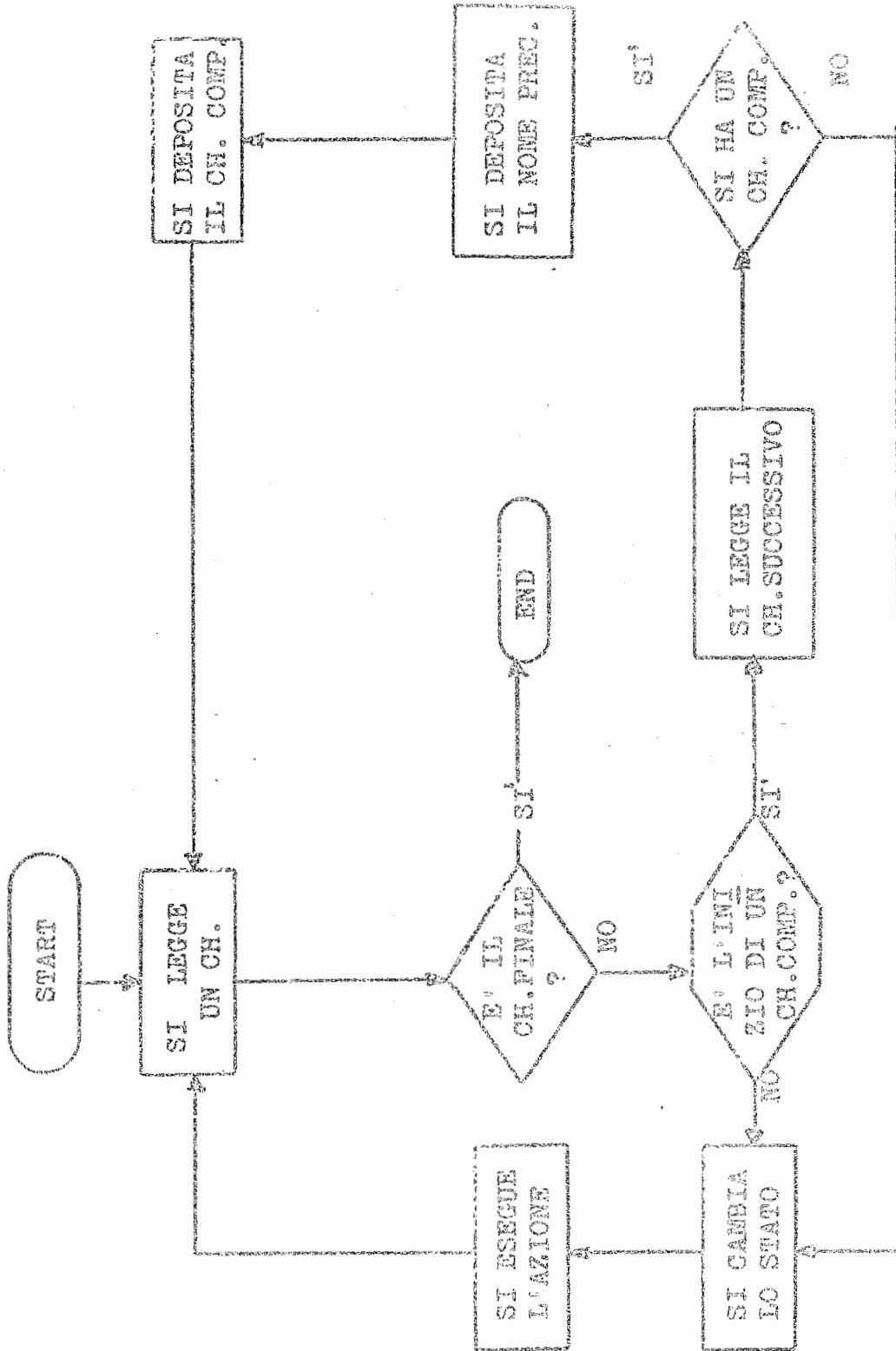
Si va così avanti fino ad incontrare uno speciale carattere (nell'implementazione attuale il CR) che segna la fine fisica del programma e che ne comanda quindi l'esecuzione.

Questa procedura è interrotta e complicata solo quando si incontra un carattere che possa essere iniziale di un <carattere composto> .

Si noti a questo proposito che i <carattere composto> possono essere diversi da quelli dati nella definizione formale del linguaggio, e possono essere anche in numero diverso; l'unica restrizione a cui devono sottostare, è che non possono consistere di più di 2 caratteri.

Il diagramma dinamico del riconoscitore è dato in figura.

DIAGRAMMA A BLOCCHI DEL RICONOSCITORE



Strettamente legato al riconoscitore é il programma che provvede alla formazione della tabella dei simboli.

Quando il riconoscitore ha isolato il nome di un oggetto, passa tale nome a questo programma che controlla se esiste già nella tabella dei simboli; se esiste già il programma si limita a inserire nella tabella di codici un puntatore alla procedura dell'oggetto; altrimenti prima di far ciò aggiunge l'oggetto stesso in coda alla tabella dei simboli. Il diagramma a blocchi di tale programma é riportato in figura.

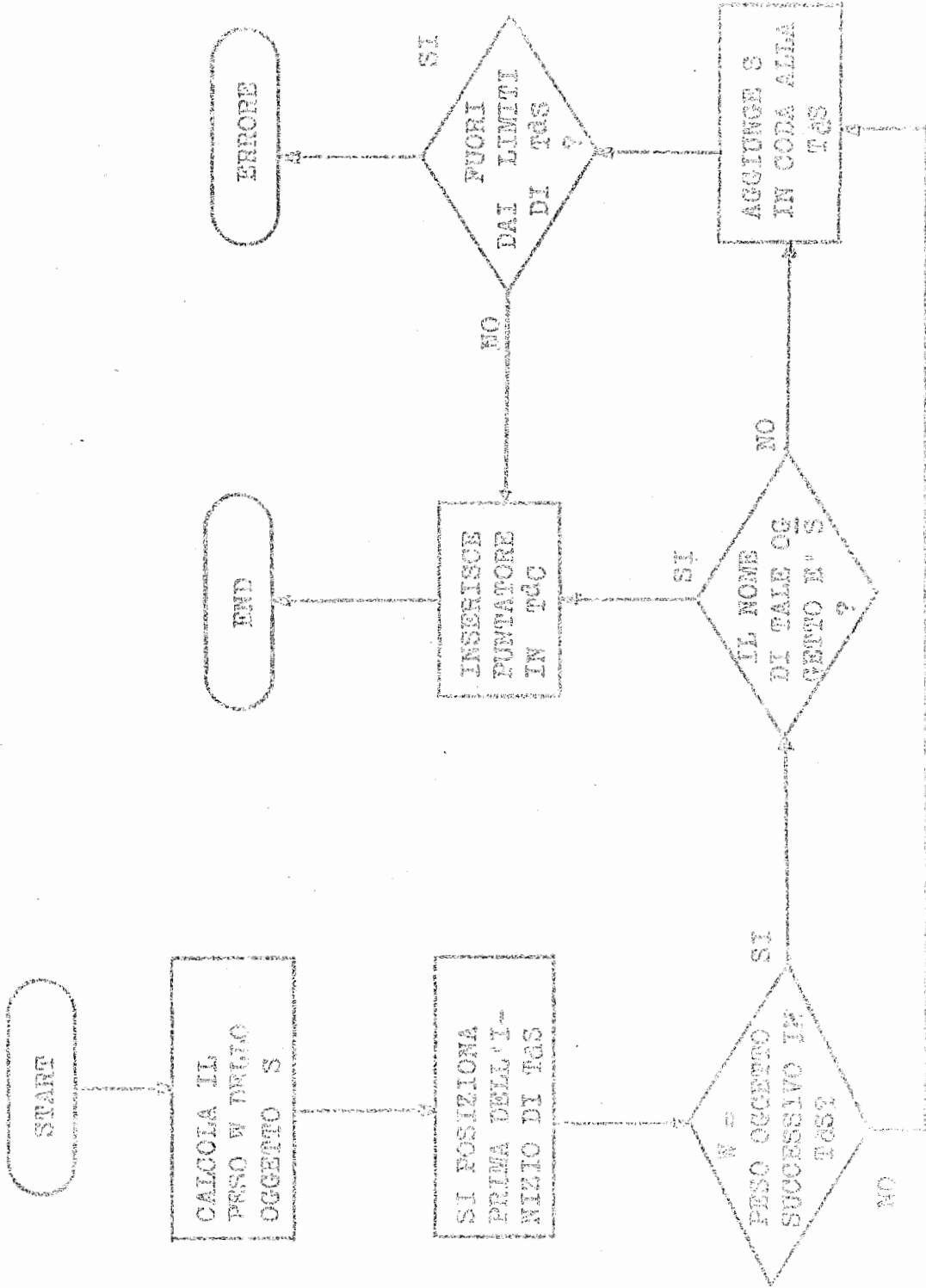
Nella velocità della ricerca ha grande importanza il peso dell'oggetto, già da noi descritte nella presentazione della tabella dei simboli. In generale, il confronto di due nomi é piuttosto laborioso, in quanto ognuno di essi occupa più celle di memoria; il peso, invece, occupando una sola cella, permette un rapido confronto e rimando al confronto diretto dei nomi solo nei (presumibilmente) rari casi che due oggetti abbiano lo stesso peso.

Nell'implementazione attuale la ricerca avviene in modo lineare; é però già previsto un metodo che faccia uso delle cosiddette tabelle hash e che permetterà un più che notevole risparmio di tempo.

PROGRAMMA DI CONTROLLO DELLA TABELLA DEI SIMBOLI

TDS = TABELLA dei SIMBOLI

TDC = TABELLA dei CODICI



Alcune caratteristiche del SEL

Come è stato accennato nella premessa, ad ogni istante il SEL si trova in uno di due stati: generativo o interpretativo. Il sistema è posto inizialmente nello stato interpretativo, e vi rimane finché non incontra il simbolo "<:" che cambia lo stato in generativo, cioè l'oggetto interno associato a questo oggetto esterno è una procedura che muta lo stato del sistema; viceversa, il simbolo ";>" riporta lo stato da generativo a interpretativo.

Il diverso comportamento del sistema a seconda che si trovi nell'uno o nell'altro stato è stato descritto nella premessa; poiché il sistema nello stato generativo genera delle procedure, si può usare questa sua caratteristica come mezzo di costruzione di oggetti interni. In effetti, quando si esegue la procedura di "<:" si inizia la generazione di un nuovo oggetto interno, che termina con l'esecuzione della procedura associata a ";>". L'oggetto interno così generato non viene di per sé assegnato ad alcun oggetto esterno; quando però si giunge ad eseguire la procedura del simbolo "->", l'ultimo oggetto interno generato viene assegnato all'oggetto esterno la cui occorrenza segue immediatamente quella di "->".

L'altro modo di definire un oggetto interno è di far uso della definizione delle macro. La procedura di "<*" definisce un nuovo oggetto interno costituito dagli oggetti le cui occorrenze seguono quella di "<*" fino a quella di "*">". L'assegnazione degli oggetti interni definiti in questo modo ad oggetti esterni, avviene con le stesse modalità descritte nel caso precedente.

Nella realizzazione effettuata si è usata la seguente organizzazione: la procedura di un oggetto il cui oggetto interno è una macro è una procedura standard che realizza la chiamata della macro stessa nel modo che stiamo per descrivere.

I puntatori (codici) che compongono il corpo della macro si trovano allo stesso posto in cui sono stati messi durante la fase di lettura del programma SEL; l'oggetto, il cui oggetto interno è la macro, ha nella sua parte riservata al valore statico un riferimento all'occorrenza del primo oggetto della macro, cioè quello che segue immediatamente "<". La procedura standard predetta assegna, in accordo a tale valore, un nuovo valore alla variabile del sistema che fa riferimento all'oggetto successivo nella sequenza di esecuzione, il che permette di eseguire la macro mediante il sistema supervisore stesso.

Si noti che alcune variabili del sistema sono state realizzate mediante vettori, il che permette di stabilire vari livelli di esecuzione; infatti, quando si passa ad eseguire una macro, il livello del sistema viene incrementato di uno, permettendo così di conservare il riferimento all'oggetto successivo, a quello che richiama la macro stessa; la procedura dell'oggetto ">" decrementa il livello di uno, il che automaticamente ristabilisce il vecchio valore delle variabili del sistema e quindi ripristina lo stato dello stesso a prima della chiamata della macro.

Tale sistema di organizzazione a livelli è usato anche per altri scopi, come per esempio, per la generazione delle etichette.

Parametri. Un'altra caratteristica del SEL è di evitare qualsiasi "scanning" non programmato; per questo il meccanismo della trasmissione dei parametri nel caso delle macro è diverso da quello consueto, che fa uso di una corrispondenza biunivoca implicita fra elementi di una lista di parametri formali e quelli della lista dei parametri attuali.

Nel SEL anche tale corrispondenza è procedurale e non dichiarativa; non esistono liste di parametri, ma esiste la possibilità di individuare un oggetto mediante un altro oggetto che ad esso faccia riferimento, ed è tale possibilità che viene usata per la trasmissione dei parametri.

Se al posto di un parametro si trova un'occorrenza di "\$REF" l'oggetto che viene identificato come parametro è quello riferito dal successore di tale occorrenza di \$REF.

Per ciò che riguarda gli oggetti base del SEL, se essi richiedono dei parametri, la loro stessa procedura controlla, indipendentemente dallo stato, se al posto di un parametro vi è un \$REF oppure un oggetto il cui oggetto interno è una macro, per individuare l'oggetto che funge da parametro attuale; nell'ultimo caso tale oggetto è il primo della macro.

In tale sistema di riferimenti hanno grande importanza i due oggetti \$PNS e \$SUC, che rappresentano un riferimento all'oggetto che si sta attualmente eseguendo e, rispettivamente, all'oggetto ad esso successivo. I valori di tali oggetti, \$PNS e \$SUC, vengono automaticamente aggiornati dal sistema.

Etichette . Il sistema permette di definire un oggetto come etichetta; l'esecuzione della procedura di un oggetto definito come etichetta realizza (genera) un salto all'occorrenza dell'oggetto successivo (cioè definito dall'oggetto \$SUC) al momento della definizione dell'oggetto etichetta come tale.

Nel caso dell'interpretazione, l'esecuzione della procedura di un oggetto etichetta consiste semplicemente nel sostituire il contenuto di una variabile del sistema, e cioè di quella che rappresenta un riferimento all'oggetto da eseguire successivamente (SUC), con il valore statico dell'oggetto etichetta, valore che viene assegnato in fase di definizione dell'oggetto come etichetta. Nel caso della generazione la cosa è più complicata perché si devono distinguere le eventualità che l'occorrenza dell'oggetto etichetta preceda o segua quella della definizione dell'etichetta stessa.

\$RET, \$REP. Nel caso dell'esecuzione di una macro definita o generata esistono come concetti primitivi la ripetizione della esecuzione dall'inizio della macro stessa ed il ritorno al punto di chiamata. A questi concetti primitivi corrisponde l'esecuzione delle procedure relative ai due oggetti base \$REP e \$RET. Poiché anche tali oggetti fanno riferimento a punti ben determinati della macro di cui fanno parte, si deve usare un meccanismo di vettori per poterli utilizzare nel caso di chiamate in cascata delle macro.

Il linguaggio conversazionale

Il SEL non é di per sé un linguaggio conversazionale, ma sfruttando le sue caratteristiche di autoestensibilità, lo si può rendere facilmente e utilmente tale. Nella versione attuale, tuttavia, questo aspetto conversazionale non é stato ottenuto facendo uso dell'autoestensibilità, ma, per ragioni di velocità di progettazione, é stato implementato direttamente.

In entrata si sono adottate le seguenti convenzioni; si é fatta innanzitutto una distinzione fra righe che contengono comandi da eseguirsi immediatamente e righe che devono essere immagazzinate in memoria per costruire un programma da eseguirsi per effetto di un comando particolare; quest'ultimo tipo di riga deve convenzionalmente iniziare col carattere "%", seguito da un numero intero, il cui valore definisce la posizione della riga nel testo del programma, di modo che le righe del programma risultino alla fine numerate in ordine crescente.

Fra gli oggetti base del SEL sono stati introdotti degli oggetti per il trattamento del testo, come per esempio la stampa, la cancellazione, ecc. del testo stesso, il che significa che questo trattamento può avvenire direttamente o mediante programmi generati, cioè é possibile introdurlo in qualsiasi autoestensione del linguaggio.

Poiché l'HP-2116B usato per l'implementazione del SEL non possiede memorie ausiliarie, il testo deve essere contenuto completamente nella memoria veloce, il che limita moltissimo l'ampiezza dei programmi. Come si é mostrato nella figura relativa all'organizzazione della memoria, si é riservata al testo un'unica pagina, il che corrisponde a 26 righe di 72 caratteri ciascuna.

Considerazioni generali

Per l'implementazione del SEL è stato usato il calcolatore HP-2116B della Scuola Normale Superiore di Pisa nell'ambito della collaborazione dell'Istituto per l'Elaborazione della Informazione con la Scuola stessa.

Il linguaggio SEL è funzionante dal Febbraio 1970 nella versione 8K qua descritta. L'implementazione ha richiesto circa ~~3000~~³⁰⁰⁰₁₀ posizioni di memoria (programmi + tabella del sistema) e 6 mesi/uomo di lavoro.

E' in corso lo studio per implementare il SEL in una versione più avanzata per ciò che riguarda la scelta degli oggetti base (resi ancora più elementari) nel calcolatore dell'I.E.I., un HP-2116B, 16K, munito di un disco. Possibilmente tale versione verrà implementata in un sistema di time-sharing.

== 000 ==