L15

Congresso A.I.C.A. 1975, Genova

ALBERTO PETTOROSSI

Istituto di Automatica dell'Università degli Studi di Roma

SULLA TERMINAZIONE IN CLASSI SUBRICORSIVE DI ALGORITMI

Abstract

In this paper we examine the termination property for programs expressed in combinatory logic as an algorithmic language. Subrecursive classes of combinators with duplicative effect are investigated and some results about existence and constructiveness of combinators without normal form are given. In the last paragraph we present some ideas toward a definition of hierarchical non terminating computations.

1. Introduzione

In questi ultimi anni sono stati sviluppati vari metodi per la verifica dei programmi e delle loro proprietà algoritmiche. Due fondamentali approcci possono essere considerati al riguardo:
i) il metodo delle asserzioni induttive introdotto da Floyd [1] e poi studiato da Manna [2];
ii) il metodo assiomatico di Hoare [3].

Questi metodi dipendono strettamente dai costrutti ammessi nel linguaggio di programmazione usato (ad esempio nell'approccio di Floyd-Manna sono accettate solo istruzioni di tipo "start", "halt", assegnazione e test), dalle modalità di specificazione delle strutture di dati e dalle operazioni su di esse [4] ed, infine, dalle proprietà che debbono essere provate.

Particolare attenzione è stata rivolta in passato alla proprietà di terminazione: se infatti un programma non termina e non produce alcun risultato, scarso è il suo effettivo interesse. In particolare sono stati già formalizzati ai fini della dimostrazione della terminazione il metodo di Floyd-Manna per i programmi "flowchart" [5] ed il metodo di Hoare per i programmi "while" [6].

Nel presente lavoro viene studiato il problema della terminazione supponendo che i programmi siano espressi nell'ambito della logica debole dei combinatori [7] considerata come un particolare linguaggio di programmazione. In questo sistema formale, però, l'assenza di tipi non permette una distinzione "a priori" tra dati e programmi e, nell'ambito dei programmi, tra primitive di controllo e costrutti. Non si può definire il concetto di grafo di flusso e vengono a perdere interesse i vari metodi di studio così come sono stati finora proposti.

D'altra parte nella logica debole dei combinatori il problema della terminazione assume una importanza particolare perchè si trasforma in quello della esistenza della cosiddetta "forma normale" ed è collegato alla determinazione del "significato" di un combinatore [8].

2. I programmi, i dati e le computazioni nella logica dei combinatori

Introduciamo ora la logica dei combinatori. Il suo alfabeto è costituito dall'insieme:
$\{I,K,S,(,),=,>,\geq, x_1,x_2,\ldots\}$ ove I,K ed S sono costanti e le x_i sono variabili.

I termini sono definiti induttivamente dalle seguenti regole:

i) una costante o una variabile è un termine;

ii) se a e b sono termini allora (ab) (cioè la combinazione applicativa di a e b) è un termine.(+)
Se a e b sono termini, a=b, a>b, a≥b sono formule.
I combinatori sono termini senza variabili.

Gli assiomi della logica dei combinatori sono:

1. $Ix_1 > x_1$ (regola di riduzione di I);
2. $Kx_1x_2 > x_1$ (regola di riduzione di K);
3. $Sx_1x_2x_3 > x_1x_3(x_2x_3)$ (regola di riduzione di S);
4. Per l'= valgono la riflessività, la simmetria e la transitività;
5. Per il > vale la riflessività;
6. Il ≥ è la chiusura transitiva di > ;
7. Per l'=, il≥ e il > vale la monotonia sinistra e destra [8];
8. Se a≥b allora a=b.

Diremo programma nella logica dei combinatori un combinatore in forma normale, in cui cioè non è possibile applicare nessuna regola di riduzione.

D'altra parte un combinatore non in forma normale può essere interpretato come un programma applicato ad uno o più dati(++). In quest'ultimo caso applicando successivamente le regole di riduzione si genera una sequenza (finita o infinita) di combinatori. Se tale sequenza è finita l'ultimo combinatore ottenuto è la forma normale (f.n.) del combinatore assegnato e costituisce il risultato del calcolo. Se la f.n. non esiste, il calcolo non termina.

Esempio 2.1

Il programma SS(KI) applicato ai dati S e K nell'ordine dà luogo a: SS(KI)SK ≡ SKK e SKK è il risultato della computazione.

Nota 2.1

Nella logica combinatoria si possono introdurre oltre ad S, K ed I delle altre costanti definite in termini di S, K ed I. Tali ad esempio sono:
B=S(KS)K; C=S(BBS)(KK); W=SS(KI) a cui corrispondono rispettivamente le seguenti regole di riduzione:
$Bx_1x_2x_3 > x_1(x_2x_3)$; $Cx_1x_2x_3 > x_1x_3x_2$; $Wx_1x_2 > x_1x_2x_2$.

Introduciamo ora alcune definizioni.

Definizione 2.1 Una sottobase B è un insieme non vuoto di combinatori indipendenti $B=\{x_1,x_2,\ldots,x_n\}$, tali cioè per cui nessuno di essi è esprimibile come combinazione applicativa degli altri [9].

Definizione 2.2 Dato un combinatore c, dicesi computazione la sequenza (finita o infinita) di combinatori da questo generata $\{c^{(0)}\equiv c, c^{(1)}, c^{(2)},\ldots\}$ per applicazione delle regole di riduzione(Δ).

Nel definire una computazione si deve pertanto specificare:
i) la struttura del combinatore, come combinazione applicativa di combinatori di una data sottobase;
ii) la cosiddetta "metaregola di computazione" riguardante il modo di ottenere il successivo combinatore della sequenza, nel caso in cui più di una regola di computazione è applicabile.

Esempio 2.2

Per il combinatore KK(WWW) possiamo avere:
KK(WWW) > KK(WWW) > ... riducendo sempre il primo W, oppure
KK(WWW) > K riducendo il primo K.

In quanto segue scegliamo come metaregola di computazione la "leftmost outermost" che assicura il raggiungimento della forma normale (se questa esiste).

3. Proprietà di classi di computazioni che non terminano

Per lo studio di classi di combinatori le cui computazioni non terminano e per l'analisi delle loro proprietà introduciamo le definizioni seguenti [7, 10].

Definizione 3.1 Un combinatore X di ordine m(∇) è proprio se

$Xx_1\ldots x_m > \chi$ ove χ è una combinazione applicativa delle variabili $x_1,x_2,\ldots,x_m$.

X ha effetto duplicativo se in χ c'è almeno una variabile x_i, $1 \leq i \leq m$, che occorre più di una volta.

X ha effetto cancellativo se esiste almeno una variabile x_i, $1 \leq i \leq m$, che non occorre in χ.

(+) Generalmente in luogo di $(\ldots((x_1x_2)x_3)\ldots x_n)$ si scrive $x_1x_2x_3\ldots x_n$.

(++) I dati a loro volta sono combinatori in forma normale oppure no. Si noti altresì che in un combinatore non in forma normale la determinazione del programma e dei dati non è in genere univoca.

(Δ) Il simbolo ≡ indica l'identità sintattica. ≠ indica la negazione di ≡.

(∇) Per la definizione di ordine si veda [7].

X ha effetto compositivo se in χ c'è almeno una applicazione che non coinvolge la variabile più a sinistra di χ.

X ha effetto permutativo se per qualche $i<j$, $1\leq i\leq m$, $1\leq j\leq m$, esiste in χ almeno una occorrenza di x_i a destra di una occorrenza di x_j.

Definizione 3.2 La chiusura applicativa di una sottobase B (indicata con B^+) è l'insieme di tutte le combinazioni applicative di χ_1.

Definizione 3.3 Dato un combinatore c, la sua computazione è ciclica con lunghezza di ciclo $n\geq 1$, se $\exists\ K\geq 0: c\geq c^{(k)} > c^{(k+1)} > \ldots\ldots > c^{(k+n)}$ ove $c^{(k+n)} \equiv c^{(k)} \neq c^{(k+i)}$ per $1\leq i\leq n-1$.

Enunciamo ora il seguente risultato generale sulla terminazione [7 pag. 180]:

Teorema 3.1 Data una sottobase P di combinatori propri, se non vi è effetto duplicativo per nessun combinatore in P allora qualsiasi combinatore in P^+ ha f.n.

In quanto segue si cerca di operare il "rovesciamento" di questo teorema indagando: i) per quali classi C_i di combinatori la presenza dell'effetto duplicativo per gli elementi di una sottobase $B\subseteq C_i$ determina l'esistenza in B^+ di un combinatore senza f.n.; ii) per quali classi C_i di combinatori si sa costruire, data una sottobase $B\subseteq C_i$, un elemento di B^+ senza f.n..

Sulla costruibilità di combinatori senza f.n. vale in generale il seguente teorema:

Teorema 3.2 Data una sottobase P di combinatori propri finita esiste un semialgoritmo per costruire un combinatore c in P^+ tale che c è senza f.n. e la sua computazione è ciclica.

Dim. Si enumerano dapprima i combinatori $\{c_i\}$ di B^+. Si calcolano poi le computazioni relative ad ognuno di essi con il procedimento descritto dalla sequenza $e_1;e_2;e_1;e_3;e_2;e_1;e_4;e_3;e_2;e_1;\ldots$ ove con il simbolo e_j si intende l'esecuzione di un passo di computo per il combinatore c_j.
Se in P^+ esiste un combinatore, diciamo c_r, con computazione ciclica e ciclo lungo n, accade per definizione che:

$$\exists\ k\geq 0: c_r \geq c_r^{(k)} > c_r^{(k+1)} > \ldots > c_r^{(k+n)} \quad \text{ove } c_r^{(k+n)} \equiv c_r^{(k)} \neq c^{(k+i)} \text{ per } 1\leq i\leq n-1.$$

Se $k=0$ allora c_r è individuabile in un "tempo" proporzionale al quadrato di r+n. Si ha infatti che la sequenza suddetta è costituita da al più $[(k+n)^2-k+n+2]/2$ elementi.

Se $k>0$ esiste $s\geq 1$: $c_s \equiv c_r^{(k)}$ poichè la sottobase considerata è costituita da combinatori propri. Il "tempo" di riconoscimento è proporzionale al quadrato di s+n. Q.E.D.

Rimanendo nell'ambito di sottobasi con combinatori propri si riesce a fornire una condizione sufficiente per l'esistenza nella chiusara della sottobase di un combinatore il quale non ha forma normale. A questo scopo diamo la:

Definizione 3.4 I sottotermini di un termine $t\in B^+$ sono definiti ricorsivamente come segue:
i) se $t\equiv(ab)$ con a e b variabili o combinatori $\in B$, allora a e b sono sottotermini di t;
ii) se $t\equiv(t_1t_2)$ ove t_1 e t_2 appartengono a B^+, allora t_1, t_2 e t sono sottotermini di t.

Enunciamo ora i seguenti risultati:

Teorema 3.3 Sia data una sottobase B a cui appartenga un combinatore proprio di ordine n tale che: $Xx_1x_2\ldots x_n > \chi$. Allora c in $\{X\}^+$ è senza forma normale se esiste in c ed in χ un sottotermine costituito da n+1 sottotermini applicati a destra.

Il sottotermine t_j si dice sottotermine (di un termine t) applicato a destra se vale:
$t \equiv t_1t_2\ldots t_j\ldots t_n$ ove $t_i, 1\leq i\leq n$, sono sottotermini di t e $t_1\in B$.

Dim. Ad ogni riduzione viene prodotto un combinatore per cui esiste la possibilità di una successiva riduzione. Q.E.D.

Corollario 3.1 Nelle stesse ipotesi del teorema 3.3 il combinatore $c\equiv \underbrace{XX\ldots X}_{n+1}$ non ha forma normale

Dim. Immediata. Q.E.D.

Una interessante sottoclasse dei combinatori propri è quella dei combinatori regolari, cioè quella dei combinatori X per cui $Xx_1x_2\ldots x_n > x_1x_1\ldots x_m$ ove le χ_i sono combinazioni applicative delle variabili $x_2,\ldots,x_n$.

Il fatto che nella regola di riduzione resta invariato il primo argomento permette sotto certe ipotesi di definire una procedura per la sintesi di combinatori senza f.n.. Valgono infatti i due seguenti teoremi:

Teorema 3.4 Sia data una sottobase B a cui appartenga un combinatore regolare X di ordine n tale che: $Xx_1\ldots x_n > x_{i_1}\ldots x_{i_\ell}x_{i_{\ell+1}}\ldots x_{i_k}$ ove $i_1=1$ e $i_\ell=i_{\ell+1}=j\geq 2$. (+).

X non abbia effetto compositivo; l'effetto cancellativo può valere solo per le variabili a sinistra

---

(+) Se j=1 allora X non è un combinatore regolare.

di x_j; l'effetto permutativo solo tra le variabili a destra di x_j.

i) Se k>n si ricade nel teorema 3.3;

ii) se k≤n allora esiste e si sa costruire un combinatore $\bar{c}$ in $\{X\}^+$ la cui computazione è ciclica.

Schema di dimostrazione. Il punto i) è immediato. Per il punto ii) trascuriamo per il momento l'eventuale presenza dell'effetto permutativo. Il combinatore $\bar{c}\equiv Xx_1x_2\ldots x_n$ in$\{X\}^+$ con computazione ciclica, costruito con la procedura che si descriverà in seguito, calcola per riduzione di tutti i combinatori basici nel sottotermine Xx_1 un combinatore $\bar{t}\equiv Xx_{k_1}x_{k_2}\ldots x_{k_p}$ con p=n tale per cui è possibile scegliere $x_1\equiv x_{k_1}$; $x_2\equiv x_{k_2}$;...; $x_n\equiv x_{k_p}$. Viene così ottenuta una computazione ciclica.

La prima delle precedenti uguaglianze può non essere imposta giacchè x_{k_1} potrebbe risultare un sottotermine di x_1 stesso. Ma la seguente procedura evita questa eventualità.

1. Scelta di un intero r>0. Produzione di un combinatore del tipo: $d\equiv Xx_{\ell_1}\ldots x_{\ell_m}$ ove $\ell_1=j$; $j\leq\ell_i\leq n$, $2\leq i\leq m$ ed m=r. Questo primo passo è possibile qualsiasi sia r>0 per iterazione del processo duplicativo con opportuna scelta di x_1.

2. I sottotermini di x_1 rimasti indeterminati al passo 1. si pongono pari a X. Si opera la loro riduzione (e questo è sempre possibile non essendoci effetto permutativo per variabili a sinistra di x_j) in modo che il combinatore d dia luogo al combinatore $\bar{t}$. ▲

Si noti che per ottenere p=n occorre in genere variare euristicamente la scelta di r.

La presenza dell'effetto permutativo per gli argomenti a destra del j-esimo lascia inalterata la validità della procedura suddetta a causa della struttura del combinatore sintetizzato. Q.E.D.

Il seguente esempio chiarirà il procedimento descritto nella dimostrazione del teorema precedente.

Esempio 3.1 Sia B={X} e $Xx_1x_2x_3x_4x_5x_6>x_1x_2x_4x_6x_5x_4$.

Passo 1. Iterazione del processo duplicativo. r=6. Se si pone $x_1\equiv Xab$ si ottiene alla successiva riduzione $abx_4x_5x_6x_4x_4$. Scegliendo poi $a\equiv Xcd$ si ha alla successiva riduzione $cdx_4x_6x_5x_4x_4x_4$.

Passo 2. I sottotermini di x_1 si pongono pari a X. Cioè: $a\equiv b\equiv c\equiv d\equiv X$. Si ha allora: $XXx_4x_6x_5x_4x_4x_4>Xx_4x_5x_4x_4x_5x_4$. Quest'ultimo combinatore può essere eguagliato a $Xx_1x_2x_3x_4x_5x_6$ scegliendo $x_1\equiv x_3\equiv x_4\equiv x_6\equiv X(Xcd)b\equiv X(XXX)X$ e $x_2\equiv x_5\equiv\bar{X}$ con $\bar{X}$ elemento generico di $\{X\}^+$. Pertanto i combinatori della forma $Xx_1\bar{X}x_1x_1\bar{X}x_1$, ove $x_1\equiv X(XXX)X$, sono elementi di $\{X\}^+$ ed hanno computazione ciclica.

Nota 3.1 Se nella procedura suddetta si prende p>n e $x_{k_1}\equiv x_{k_p}$ per p+1≤i≤n allora il combinatore $\bar{t}$ ottenuto è senza forma normale ma non determina una computazione ciclica.

Il risultato del teorema precedente può essere esteso. Si può stabilire infatti il teorema seguente:

Teorema 3.5 Il teorema 3.4 vale supponendo la presenza dell'effetto permutativo tra le variabili x_{i_r} , ove $2\leq r\leq\ell-1$.

Dim. La procedura di cui al teorema 3.4 potrebbe cadere in difetto se dovesse essere imposta la uguaglianza tra x_1 in $\bar{c}$ ed un sottotermine di x_1 (diverso da x_1 stesso) in $\bar{t}$. Ma per la stessa presenza dell'effetto permutativo esiste in $\bar{t}$, a sinistra di x_j, almeno una variabile x_i con $2\leq i\leq\ell-1$ da poter uguagliare a x_1. Q.E.D.

Esempio 3.2 Sia B={X}e $Xx_1x_2x_3x_4x_5>x_1x_3x_2x_5x_5$. Scegliendo $x_1\equiv Xab$ si ha: $x_1x_3x_2x_5x_5>ax_3bx_5x_5x_5$ che può essere uguagliato a $Xx_1x_2x_3x_4x_5$ ponendo $a\equiv X$; $x_1\equiv x_3\equiv x_4\equiv x_5\equiv XX\bar{X}$ e $b\equiv x_2\equiv\bar{X}$ è un elemento generico di $\{X\}^+$. Da cui il combinatore $\bar{t}\equiv X(XXX)\bar{X}XXX$ ha computazione ciclica con lunghezza di ciclo 2 ($x_1\equiv X$).

Un altro promettente approccio allo studio della terminazione nella logica combinatoria è quello di determinare in che modo le limitazioni delle "risorse" (numero di riduzioni, lunghezza dei combinatori, ecc.) siano legate alla possibilità di stabilire l'esistenza e la costruibilità di combinatori senza f.n. In questa prospettiva si inquadrano i risultati seguenti.

Definizione 3.5 La lunghezza ℓ di un combinatore c in B^+ è definita ricorsivamente come segue:

i) se c è un elemento di B, $\ell(c)=1$;

ii) se $c=(c_1c_2)$, $\ell(c)=\ell(c_1)+\ell(c_2)$.

Fatto 3.1 Sia data la sottobase B={X} ove X è un combinatore proprio. Se nella computazione relativa a $c\in\{X\}^+$ esistono almeno m combinatori di lunghezza n+1 e $m>a_{n,n}$ ove $a_{i,j}=a_{i,j-1}+a_{i-1,j}$ con $a_{i,1}=1$ per ogni i>0; $a_{i,2}=i$ e $a_{i,i+1}=a_{i,i}$, allora la computazione di c è ciclica.

Dim. Il valore $a_{n,n}$ è il numero di combinatori distinti in $\{X\}^+$ di lunghezza n+1 (cioè il numero di b-alberi distinti con n+1 foglie). Se nella computazione occorrono più di $a_{n,n}$ combinatori di lunghezza n+1 almeno uno di essi comparirà due volte. Q.E.D.

Teorema 3.6 Sia dato un combinatore regolare X con effetto duplicativo tale che: $Xx_1x_2\ldots x_j\ldots x_k > x_1x_2\ldots x_jx_j\ldots x_k$ ove $k\geq 2$. Se per le prime n riduzioni di $c\in\{X\}^+$ si ottengono combinatori lunghi n+1 allora la computazione di c è ciclica ed i combinatori di tale computazione sono tutti lunghi n+1.

Dim. Dimostreremo il teorema per induzione sulla lunghezza dei combinatori. Il teorema è vero per i combinatori lunghi 3. Proviamo ora che se è vero per quelli lunghi ℓ con $3\leq\ell\leq n$, è vero anche per quelli lunghi n+1.

Sia $\bar{c}$ un combinatore lungo n+1. Se $\bar{c}\equiv Xc_1c_2\ldots c_jc_{j+1}\ldots c_k\ldots c_p$ alla prima riduzione dà luogo a $c_1c_2\ldots c_jc_jc_{j+1}\ldots c_k\ldots c_p$ che è lungo n+1 se e solo se $c_j\equiv X$. Ora se $k<p$ si applica l'ipotesi induttiva poichè al procedere della computazione questa non interessa più le variabili $c_{k+1},\ldots,c_p$ che si trovano a destra del k-esimo argomento. Se non si applica l'ipotesi induttiva in nessuno dei primi n passi, il numero di sottotermini applicati a destra deve essere rimasto invariato per i primi n combinatori della computazione. Questo è possibile solo se $\bar{c}\equiv\underbrace{XX\ldots X}_{n+1}$ per cui, però, il teorema è banalmente vero. Q.E.D.

Prima di chiudere il paragrafo vogliamo verificare che è falsa la congettura secondo cui se in una sottobase B esiste un combinatore proprio con effetto duplicativo, allora in B^+ esiste un combinatore senza f.n..(+)

Fatto 3.2 Sia data B={X}ove $Xx_1x_2\ldots x_n > x_1x_1$. Allora se $n\geq 2$ qualsiasi elemento in B^+ ha f.n..

Dim. Sia $c\equiv Xx_1\ldots x_p$ elemento di B^+. Se $p<n$ allora c è in f.n.. Se $p\geq n$ può essere: i) $c\equiv\underbrace{XX\ldots X}_{p}$ e allora c ha f.n.; ii) in un sottotermine x_i, $1\leq i\leq p$ deve esserci almeno un sottotermine XX. Quando si riduce quel sottotermine XX allora si determina una progressiva riduzione del numero dei sottotermini dei combinatori della computazione. Q.E.D.

Nota 3.2 Se si suppone n=1 si cade nelle condizioni del teorema 3.3.

4. Verso una gerarchia di classi di combinatori senza forma normale

In base ai risultati del paragrafo precedente scaturisce che ai fini della esistenza e della costruibilità di combinatori senza forma normale l'effetto duplicativo ha un ruolo fondamentale. Ed è per questo che abbiamo indagato sui suoi rapporti con gli altri effetti, ma nell'indagine fin qui svolta non è stato ancora preso in considerazione l'effetto compositivo. La presenza di quest'ultimo effetto può far sì che l'effetto duplicativo non sia più, sotto certe ipotesi, una condizione sufficiente per l'esistenza della forma normale. Vale infatti:

Fatto 4.1 Sia data la sottobase B={X} ove $Xx_1x_2\ x_1(x_1x_2)$. Tutti i combinatori in $\{X\}^+$ hanno f.n.

Dim. Il combinatore X è il combinatore 2 [11] e ogni combinazione di soli 2 ha forma normale poichè rappresenta un numero standard [7]. Q.E.D.

Comunque l'effetto compositivo non riesce sempre ad "inibire" l'effetto duplicativo nel senso che è possibile trovare una sottobase con un combinatore con effetto duplicativo e insieme compositivo nella cui chiusura ci sia un combinatore senza forma normale. Possiamo infatti enunciare il seguente teorema.

Teorema 4.1 Sia data la sottobase B={X} ove $Xx_1x_2\ldots x_{n-1}x_n > x_1x_2\ldots x_{n-1}(x_nx_n)$ con $n\geq 2$. Allora: i) esiste (e c'è un algoritmo per costruirlo) un combinatore c in $\{X\}^+$ che non ha f.n.; ii) nessun elemento in $\{X\}^+$ ha computazione ciclica.

Dim. i) Per opportuna scelta di x_1 si riesce a costruire, qualsiasi sia k, un combinatore c con il sottotermine s_k, ove s_k è definito ricorsivamente come segue:
$s_0\equiv X$; $s_k\equiv Xs_0s_1\ldots s_{k-1}$. Si ottiene un combinatore con computazione infinita facendo sì che k sia

---

(+) Questo fatto è stato provato indipendentemente in [10].

pari ad n.

ii) Ad ogni passo la lunghezza cresce o resta invariata. Può restare invariata, permettendo così una computazione ciclica, solo se l'n-esimo argomento è sempre X. E questo è impossibile.

Q.E.D.

Si potrebbe pensare che le cose possano andare diversamente se l'effetto compositivo e duplicativo interagissero in maniera, diciamo così, meno semplice. Ma proprietà analoghe a quelle viste nel teorema precedente valgono per la sottobase {S}, ove l'effetto compositivo non interessa solo le variabili duplicate.

Fatto 4.2 i) Esiste un combinatore in $\{S\}^+$ che non ha f.n. ii) Non esiste in $\{S\}^+$ un combinatore che determina una computazione ciclica.

Dim. Punto i). Sia 3≡SSS. Allora vale 3ab>ab(Sab). Ebbene il combinatore s≡S33(S33) non ha f.n. Sia infatti $\{s_i\}$ per $i\geq 0$ la computazione di $s\equiv s_0$ e sia $\{n_i\}$ la sequenza di interi tale che $n_1=0$ e $n_i=n_{i-1}+i+1$. Se indichiamo i primi quattro sottotermini applicati a destra di s_{n_j} con $S33s'_{n_j}$ si può facilmente verificare che: $s'_{n_{j+1}} \equiv 3s'_{n_j}$.

Punto ii). Supponiamo per assurdo che esista un tale combinatore χ. Poichè la regola di riduzione è $Sx_1x_2x_3>x_1x_3(x_2x_3)$ ad ogni passo di computazione la lunghezza del combinatore ottenuto aumenta a meno che non sia $\ell(x_3)=1$ (nel qual caso resta costante). Affinchè sia possibile una computazione ciclica occorre pertanto che ad ogni passo $\ell(x_3)$ sia pari ad 1. Il che è assurdo a causa della presenza dell'effetto compositivo. Q.E.D.

In relazione a questi risultati può essere sviluppata una ipotesi di stratificazione di classi di combinatori attraverso una gerarchia di complessità crescente secondo le seguenti linee.

Si considera la classe di tutte le sottobasi costituite da un solo combinatore proprio e si pongono ad un livello più basso le sottobasi con il solo effetto duplicativo, mentre ad un livello più alto si pongono quelle in cui è presente anche l'effetto compositivo. Questa stratificazione risulta congruente con una gerarchia di classi di computazioni per lunghezza di ciclo crescente, supponendo di ciclo infinito le computazioni infinite non cicliche (si vedano al riguardo il teorema 4.1 ed il fatto 4.2). I risultati sulla costruibilità di cui ai teoremi 3.5, 3.6 e 4.1, confortano questa ipotesi di stratificazione. Si nota infatti che al livello più basso della gerarchia è sempre possibile ottenere, nella chiusura di una sottobase, una computazione infinita non ciclica a partire da una computazione ciclica, mentre al livello più alto ciò non si verifica, essendo possibile che non esista neppure una computazione ciclica.

Nell'ambito poi delle computazioni cicliche, una più raffinata stratificazione di classi può essere fatta in base alla complessità strutturale (lunghezza, profondità parentetica, ecc. [9]) del sottotermine per la cui riduzione si completa un ciclo della computazione (si vedano i teoremi 3.4 e 3.5 ove si ottiene un ciclo della computazione per riduzione dei primi due sottotermini applicati a destra).

D'altra parte, se si ammette una classificazione dei combinatori per effetti (duplicativo, cancellativo, compositivo e permutativo), diventerebbe poco significativa una gerarchizzazione in base alla struttura dei combinatori appartenenti ad un ciclo della computazione. Si è visto infatti (esempi 3.1 e 3.2) che può esser possibile, dato un combinatore X, sintetizzare tutta una classe di elementi in $\{X\}^+$ la cui computazione è ciclica, con la stessa lunghezza di ciclo, ma la cui struttura è arbitraria, almeno entro ampi limiti.

5. Conclusioni

Nel presente lavoro abbiamo studiato la terminazione dei programmi ed i problemi ad essa collegati, supponendo di usare come linguaggio di programmazione quello associabile ad un particolare sistema formale: la logica combinatoria.

In questo caso il problema della terminazione si trasforma in quello della esistenza della "forma normale" dei combinatori.

Sono state introdotte le nozioni di "programma" (come combinatore in forma normale), di "dato" e di

"computazione" e sono state considerate varie classi subricorsive di combinatori facendo uso del concetto di sottobase [9]. Per alcune di queste classi abbiamo fornito condizioni di esistenza e costruibilità di combinatori non in forma normale, a cui corrispondono computazioni infinite (cicliche e non).

Altri risultati sono stati ottenuti utilizzando misure di complessità strutturale e computazionale introdotte in [9] ed indagando sulla relazione di una loro limitazione con l'esistenza di computazioni cicliche.

Sono state svolte infine alcune considerazioni sulle modalità secondo cui definire una gerarchia di complessità crescente per classi di combinatori senza forma normale.

BIBLIOGRAFIA

[1] Floyd, R.W. *Assigning Meaning to Programs* Proc. Symp. Appl. Math.,19 (1967).

[2] Manna, Z. *The Correctness of Programs* J.C.S.S., 3 (2) pp.119-127 (1969).

[3] Hoare, C.A.R. *An Axiomatic Basis of Computer Programming* C.A.C.M., 12 (10) pp.576-580 (1969).

[4] Liskov, B. and S. Zilles *Specification Thechniques for Data Abstractions* Proc. of International Conference on Reliable Software (april 1975).

[5] Manna, Z. *Termination of Programs Represented as Interpreted Graphs* S.J.C.C. pp.83-89 (1970).

[6] Manna, Z. and A. Pnueli *Axiomatic Approach to Total Correctness* Acta Informatica 3 pp.243-263 (1974).

[7] Curry, H.B. and R. Feys *Combinatory Logic* Vol. 1 North Holland (1968).

[8] Böhm, C. *Alcune proprietà delle forme β-η-normali del λ-K-calcolo* Pubblicazioni I.A.C. N.696 C.N.R. Roma (1968).

[9] Batini, C. e A. Pettorossi *On Subrecursiveness in Weak Combinatory Logic* Proc. of the Symp. on λ-calculus and Computer Science Theory. Rome (1975).

[10] Böhm,C. *Private Communication* (1975).

[11] Böhm,C. *Introduction to CUch* Ed. Caianiello (1966).