47

Computer Sciences Department
1210 West Dayton Street
Madison, Wisconsin 53706

PROOFS OF ALGORITHMS FOR ASYMPTOTIC SERIES[1]

by

Ralph L. London[2]
and
John H. Halton[2]

Technical Report #54A

May 1969

[1] This research was supported by the National Science Foundation under Grants GP 7069 and GJ 171, and by the Mathematics Research Center, United States Army under Contract Number DA-31-124-ARO-D-462.

[2] Computer Sciences Department and Mathematics Research Center, United States Army.

ABSTRACT

Six algorithms for computing asymptotic series as formulas are proved to be correct.

Acknowledgement

We wish to thank Mrs. Laura Collins for her usual careful typing of the manuscript (despite our efforts to render it illegible and confusing by a mass of corrections, afterthoughts, and editorial rearrangements), and especially for her patient and expert arrangement and execution of the flowcharts, using mainly a typewriter--a machine that was never intended to produce such copy.

PROOFS OF ALGORITHMS FOR ASYMPTOTIC SERIES

by

Ralph L. London
and
John H. Halton

Introduction

In a previous paper [1], several algorithms appear, intended as part of a proposed package in a formula-manipulation system, to compute and to manipulate asymptotic series. The purpose of the present paper is to prove that these algorithms meet their specifications and to thereby remove lingering doubts about their correctness. The algorithms are not complex (with the possible exception of the last one), and once they are "understood," it is easy to "convince oneself" that they are correct. Nevertheless, formal proofs of their correctness are quite appropriate since they firmly establish the correctness of the algorithms, or in other words, certify the algorithms in a new way. The question of the algorithms' being correct reduces to the question of whether the proofs are correct--certainly a significant advance. The question of the correctness of proofs is, of course, the same issue that arises with all proofs, and the point is that the present proofs are, of necessity, subject to that question.

The present paper assumes the contents of [1]. (For the benefit of readers unfamiliar with [1], we here include an Appendix containing a minimum of essential prerequisites, including the algorithms.) Each of the six algorithms to be proved calculates the first n non-zero terms of an asymptotic series. The integer n is an input parameter with the assumption that $n \geq 1$. Sufficient terms are assumed to exist both for the answer and for any series needed in computing the answer. In other words, there are no program checks or error exits to guard against this contingency although they easily could be added (and proved). It is clearly true but not proved here that all input series are unaltered by computations involving them.

Since the answer is computed as a formula and not as a numeric quantity, round-off problems do not exist. The proofs assume that all formulas can be computed with no space problems, and the integer variables are assumed small enough to cause no overflow.

In order to facilitate the presentation of the proofs, each algorithm is restated in a flowchart which is identical to the corresponding algorithm in [1]. No simplifications or alterations have been made either to the quantities computed or in the interconnection of the statements. The only "difference" is in the placement of the boxes on the page to avoid crossing of lines. While the proofs apply to the flowcharts, it is clear by simply checking the flowchart against the algorithm, that the proofs also apply to the algorithms in their original representation.

The proofs of correctness are given by a technique described, with examples, in both [2] and [3]. The idea is that assertions concerning the progress of the computation are made between boxes of the flowchart, and the proof consists of demonstrating that each assertion is true each time control passes that assertion, under the assumption that the previously encountered assertions are true. It is essentially an inductive technique that shows that there is no first false assertion. Termination of the algorithm is shown separately. Also exploited are strategies of proof which are discussed in [4], such as the table of variable-value changes and sectioning.

Each box of a flowchart is given an integer number in order that the proof may refer to that box. The assertions are numbered "n.m" with "n" usually the number of the box preceding the assertion and with "m" used to distinguish several assertions made at the same point. (See, for example, box 2 and assertions 2.1 and 2.2 in the flowchart for algorithm (91) [numbers in parentheses refer to equations in [1]; notation such as (*1) refers to an equation in the present report].) The variables of the assertions are the same as those of the algorithm except for dummy variables which must be new.

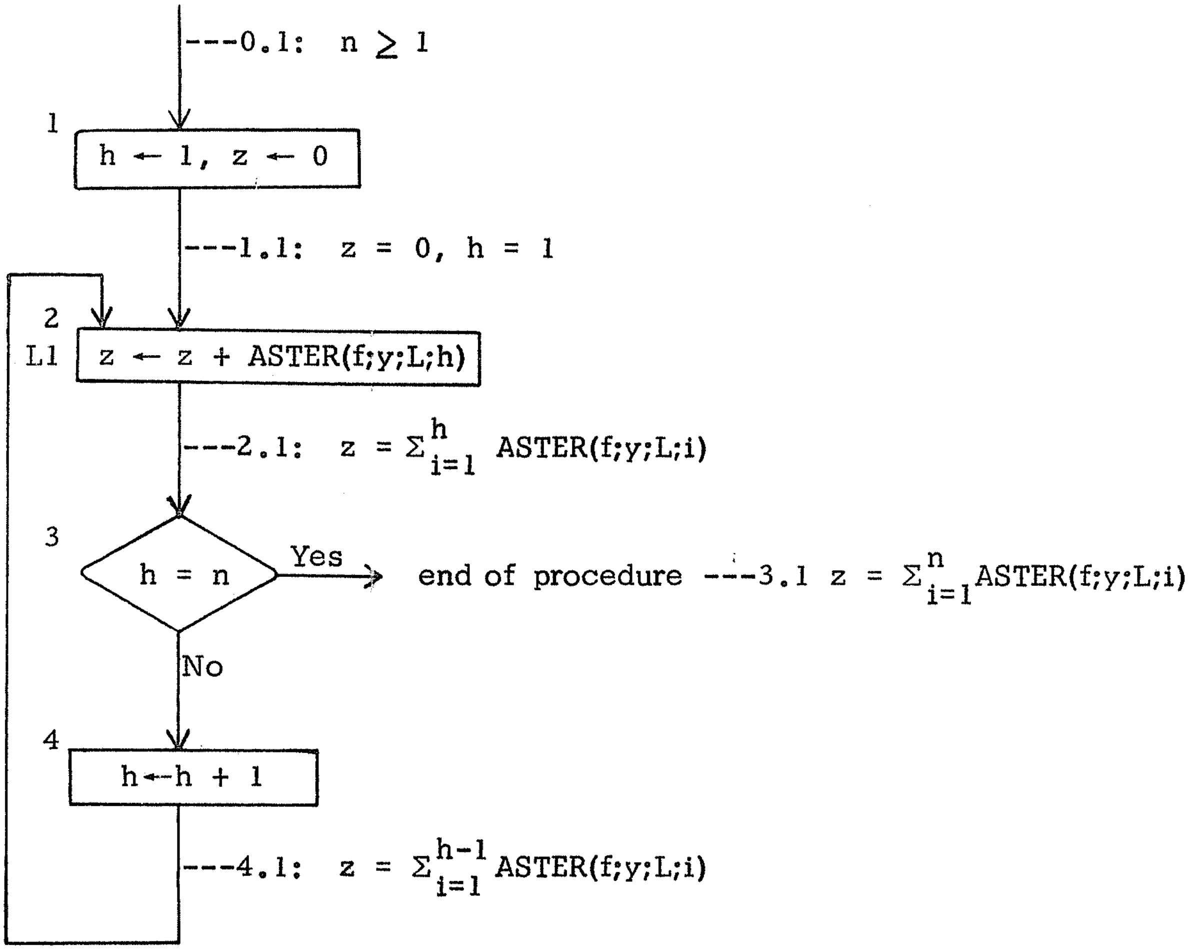
The symbols "//" are used as an end of proof sign. We now turn to a statement of the processing claimed for each algorithm and a proof that the claim is realized.

Proofs of the algorithms

The first algorithm to be proved is (97). While a complete proof of this obvious algorithm is probably unnecessary, one is still given to illustrate both the proof technique and the notation used in a proof.

Theorem: Algorithm (97) computes $z = \Sigma_{i=1}^{n}$ ASTER(f;y;L;i).

Proof: The following flowchart represents (97), and it has the assertions added.


---0.1: $n \geq 1$
1
$h \leftarrow 1,\ z \leftarrow 0$
---1.1: $z = 0,\ h = 1$
2
L1
$z \leftarrow z +$ ASTER(f;y;L;h)
---2.1: $z = \Sigma_{i=1}^{h}$ ASTER(f;y;L;i)
3
$h = n$
Yes
end of procedure ---3.1 $z = \Sigma_{i=1}^{n}$ASTER(f;y;L;i)
No
4
$h \leftarrow h + 1$
---4.1: $z = \Sigma_{i=1}^{h-1}$ASTER(f;y;L;i)

Flowchart for Algorithm (97)

Reasons for the truth of each assertion follow: (Since assertion 2.1 can be reached from either assertion 1.1 or 4.1, it is necessary to consider both possibilities.)

0.1: Assumption on n.

1.1: By 1.

2.1: From 1.1: By 1.1 and 2, $z = ASTER(f;y;L;1)$. From 4.1: By 4.1 and 2.

3.1: By 3, $h = n$. Hence 3.1 is 2.1 with $h = n$.

4.1: By 2.1 with change in h at 4.

(97) terminates since $n \geq 1$, h starts at 1, and h is stepped by 1 (after testing); whence h must eventually equal the unchanged n, and therefore the end of the procedure is reached. The desired result of (97) is assertion 3.1.//

Algorithm (91) is more complicated to prove and requires additional notation for use in the assertions. If $s \geq 1$, define $z = A(s)$ to mean

$$z = \Sigma_{r=1}^{s} a_{j_r} \varphi_{j_r} \quad \text{where} \tag{*1}$$

$$\left.\begin{array}{l} a_{j_r} \neq 0 \quad \text{for} \quad r = 1,2,\ldots,s \ (0 \leq j_1 < j_2 < \ldots < j_s), \\ a_t = 0 \quad \text{if} \quad 0 \leq t < j_s \quad \text{and} \quad t \neq j_r \ (r = 1,2,\ldots,s). \end{array}\right\} \tag{*2}$$

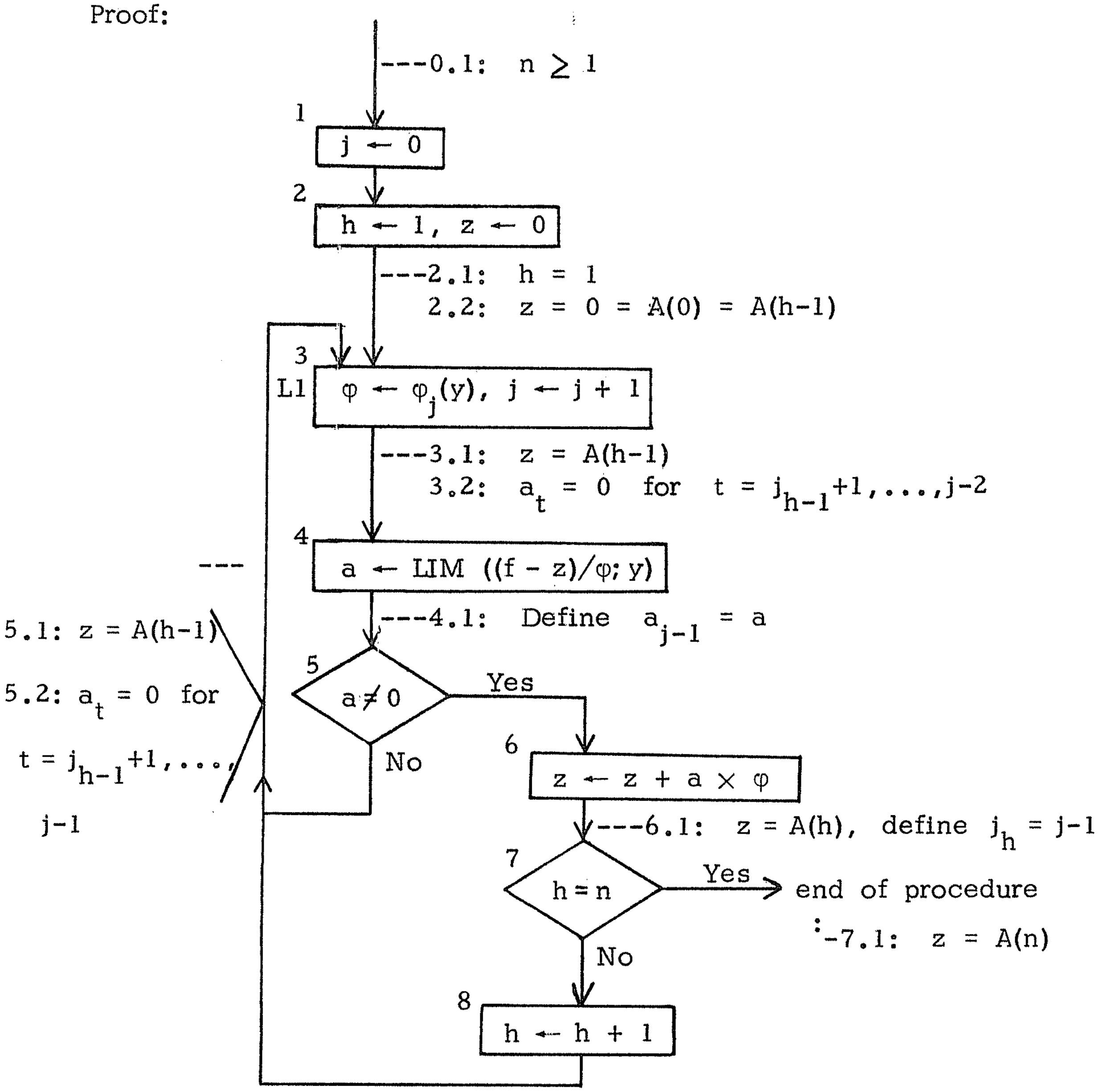
If $s = 0$, define $A(0) = 0$. $A(s)$ is (80) and the conditions (79) but with different variables, and this is to avoid conflict with the variables used in (91). Using the $A(s)$ notation, (91) is to compute $z = A(n)$ which is assertion 7.1 in the flowchart.

In order to be able to make the necessary assertions about the coefficients as required by $A(s)$, each quantity a that is computed by box 4 will be denoted by a_{j-1}, using the current value of j, as shown in the flowchart at assertion 4.1. Similarly for each non-zero coefficient a, assertion 6.1 defines the index j_h $(h \geq 1)$ needed to be able to refer to the non-zero coefficients. Define $j_0 = -1$.

The INITIALIZE L and CALL L statements in [1] have been replaced by the respective "Note" since that is the effect of the use of L.

Theorem: Algorithm (91) computes $z = A(n)$.

Proof:



---0.1: $n \geq 1$
1
$j \leftarrow 0$
2
$h \leftarrow 1,\ z \leftarrow 0$
---2.1: $h = 1$
2.2: $z = 0 = A(0) = A(h-1)$
3
L1
$\varphi \leftarrow \varphi_j(y),\ j \leftarrow j + 1$
---3.1: $z = A(h-1)$
3.2: $a_t = 0$ for $t = j_{h-1}+1,\dots,j-2$
4
$a \leftarrow \mathrm{LIM}\ ((f - z)/\varphi;\ y)$

---4.1: Define $a_{j-1} = a$
5.1: $z = A(h-1)$
5.2: $a_t = 0$ for $t = j_{h-1}+1,\dots,j-1$
5
$a \neq 0$
Yes
No
6
$z \leftarrow z + a \times \varphi$
---6.1: $z = A(h)$, define $j_h = j-1$
7
$h = n$
Yes
end of procedure
-7.1: $z = A(n)$
No
8
$h \leftarrow h + 1$

Flowchart for Algorithm (91)

Reasons for the truth of each assertion follow:

0.1: Assumption on n.

2.1-2.2: By 2 and definition of A(0).

3.1: From 2.1: By 2.2. From 5.1: By 5.1.

3.2: From 2.1: Since $j_0 = -1$, and $j = 1$ by 1 and 3, the range of t is null. From 5.1: By 5.2 with change in j at 3.

5.1: From 5 directly: By 3.1 and 5. From 8: By 6.1 with change in h at 8.

5.2: From 5 directly: $a = a_{j-1} = 0$ by 5, so 3.2 also holds for $t = j-1$ by 4.1. From 8: The range of t is null, since $j-1 = j_h$ (at 6.1) $= j_{h-1}$ (at 5.2) by 6.1 and 8.

6.1: Assuming box 4 does give the desired coefficient, 6.1 follows from 3.1, 3.2, 5($a \neq 0$), and 6.

7.1: By 7, $h = n$. Hence 7.1 is 6.1 with $h = n$.

(91) will terminate provided h reaches n. Since h starts at 1 and is stepped by 1, h will equal the fixed $n \geq 1$ unless h stays fixed, i.e., unless the path from boxes 3 to 5 and (directly) back to 3 is taken infinitely often. This corresponds to an infinite succession of zero coefficients, a possibility that is ruled out by the first paragraph on page 32 of [1].//

Algorithm (100) for computing the sum of asymptotic series is proved next and, as a corollary, algorithm (99) is shown to be correct. ((100) is (99) suitably modified.) To compute (99) Lemma 12(i) is used, i.e., formula (50) is computed. The inner sum from $i = 1$ to k reflects the collecting of like φ_j terms into a single term. Hence each coefficient a_j in the asymptotic sum is

$$a_j = \Sigma_{u=1}^{k} c_u a_{uj} \ . \tag{*3}$$

If φ_j does not appear in f_i, then the corresponding a_{ij} is zero.

In (99) $k \geq 1$ is the number of f_i's being summed, so each a_j consists of k summands, counting zeros. In (100), which uses (51), the indices k_h $(h \geq 0)$ provide an upper bound on the number of f_i's which must be checked to insure that all terms involving φ_h have been included in the computation of $\Sigma_{i=1}^{\infty} c_i a_{ij}$, as explained in the Appendix. Hence in (100), a suitable k for the analog of (*3) is

$$k = \max_{-1 \leq h \leq j} (k_h) \tag{*4}$$

where, to simplify the notation, k_{-1} is defined as $\max(1, k_0)$ thus guaranteeing that $k \geq 1$ for all k. This bound on k is needed to insure termination.

The quantities h_i and ℓ_i in (100) are used to record the first term in f_i that has not already been included in the computation of the result. h_i gives the term number in f_i with only non-zero terms being counted, starting at 1; and ℓ_i gives the index of the h_ith φ according to the original numbering of the φ sequence, starting at 0. The notation $B(s)$ will mean that h_r and ℓ_r for $r = 1,2,\ldots,s$ are all set with the above meaning.

Assertions such as 3.2 and 5.2 which appear in the proof of (91) have been omitted for brevity in the proof of (100) and in subsequent proofs. In all cases the details are the same as in (91). Furthermore, since all uses of ASCOF, ASTER and ASIND have certain parameters identical, only the first and last parameters are written.

The assertions for this proof and for subsequent proofs do not appear on the flowchart because of space problems. Instead notation such as "3.1" or "7.1-7.4" is placed where the assertions belong, and the actual assertions appear below the flowchart.

Theorem: Algorithm (100) computes z of operation (90) with $f = \Sigma_{i=1}^{\infty} c_i f_i$ assuming the existence of the indices k_h as described on page 34 of [1] and in the Appendix.

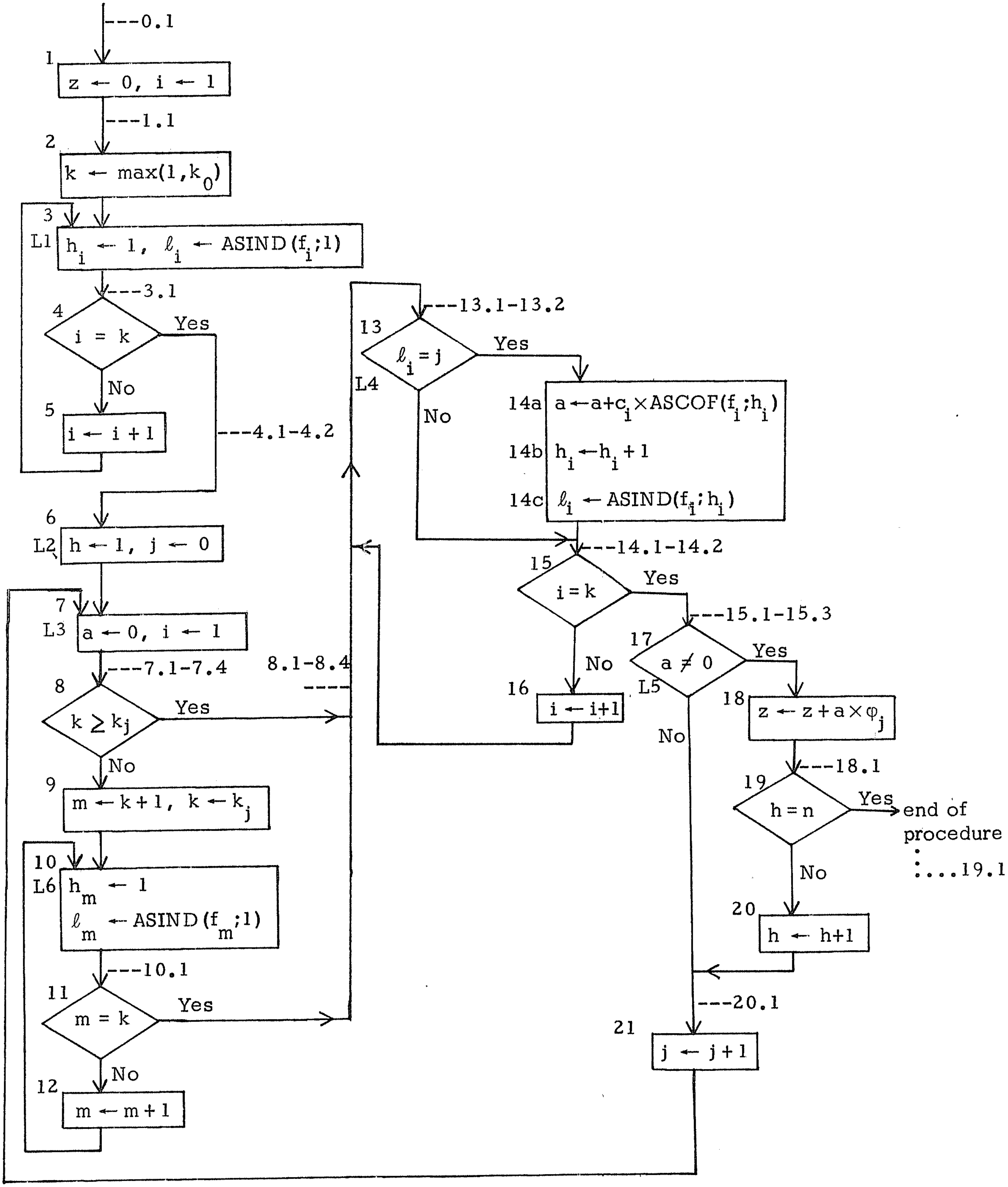
Lemma: Variables in (100) are changed in value only as follows:

Variable	Changed at Box Number
a	7, 14a
a_j	defined at assertion 15.3
c_i	--
h	6, 20
h_i	3, 10, 14b
j	6, 21
i	1, 5, 7, 16
k	2, 9
k_h	--
ℓ_i	3, 10, 14c
m	9, 12
n	--
z	1, 18

Proof: Inspection of the flowchart.//

This table is useful in verifying that assertions remain unchanged, especially over several boxes. The notation "--" emphasizes that the variable is unchanged throughout.

Proof of theorem:



Flowchart for Algorithm (100)

Below appear the assertion numbers, the actual assertions, and the reasons why each assertion is true.

0.1: $n \geq 1$: Assumption on n.

1.1: $z = 0$, $i = 1$: By 1.

3.1: $B(i)$: From 1.1: $i = 1$ so $B(1)$ by 3. From 5: By 3.1, 5, and 3.

4.1: $B(k)$: By 3.1 with $i=k$. By 2, assertion 4.1 implies $B(k_0)$ and $B(1)$.

4.2: $z = 0 = A(0)$: z is unchanged from 1.1.

7.1: $z = A(h-1)$: From 4.1: By 4.2 since $h = 1$ at 6. From 20.1: By 20.1.

7.2: $B(k)$: From 4.1: Unchanged from 4.1. From 20.1: Unchanged from 15.1.

7.3: $a = 0$, $i = 1$: By 7.

7.4: $k = \max\limits_{-1 \leq v \leq j-1} (k_v)$: Recall that $k_{-1} = \max(1, k_0)$. From 4.1: By 6, $j = 0$, and by 2, $k = k_{-1}$. From 20.1: By 8.4 with change in j at 21.

8.1: $B(k)$: From 8 directly: By 7.2. From 11: By 10.1 with $m = k$.

8.2: $a = 0$, $i = 1$: Unchanged from 7.3.

8.3: $z = A(h-1)$: Unchanged from 7.1.

8.4: $k = \max\limits_{-1 \leq v \leq j} (k_v)$: From 8 directly: By 7.4 and the test at 8. Note by 2 and 8 that 8.4 is always reached directly from 8 if $j = 0$. From 11: By 7.4 (with $j > 0$), 8, and 9 ($k \leftarrow k_j$).

10.1: B(m): From 9: By 7.2, 9(m = k+1), 10, and $k < k_j$ at 8. From 12: By 10.1, 12, and 10.

13.1: B(k): From 8.1: By 8.1. From 16: Unchanged from 14.1.

13.2: $a = \Sigma_{u=1}^{i-1} c_u \text{ASCOF}(f_u;h_u)\, \delta_{j\ell_u}$, i.e., the term c_u ASCOF $(f_u;h_u)$ appears iff $j = \ell_u$: From 8.1: By 8.2, vacuous sum. From 16: By 14.2 with change in i at 16.

14.1: B(k): From 14: By 13.1, 14b, and 14c, since after 14a only both h_i and ℓ_i must be reset. From 13 directly: Unchanged from 13.1.

14.2: $a = \Sigma_{u=1}^{i} c_u \text{ASCOF}(f_u;h_u)\, \delta_{j\ell_u}$: From 14: By 13.1, 13.2, and 14a. From 13 directly: By 13.1 and the test at 13 ($\ell_i \neq j$) (means φ_j does not appear in f_i).

15.1: B(k): By 14.1.

15.2: z = A(h-1): Unchanged from 8.3.

15.3: $a = \Sigma_{u=1}^{k} c_u \text{ASCOF}(f_u;h_u)\, \delta_{j\ell_u}$, define $a_j = a$ by (*3): By 14.2 with i = k. By 8.4, the value of k is as required for (*4).

18.1: z = A(h): By 15.2, 15.3, and $a \neq 0$ at 17.

19.1: z = A(n): By 18.1 with h = n.

20.1: z = A(h-1): From 20: By 18.1 with change in h at 20. From 17 directly: By 15.1, and a = 0 at 17.

Termination for (100) follows as in (91). The path for unchanging h is from boxes 7 to 17 to 21 to 7. The path from 7 must reach 17 in a finite number of steps. The path from 7 to 7 that avoids box

20 infinitely often is ruled out exactly as it is excluded in (91). Box 2 and assertions 7.4 and 8.4 insure $k \geq 1$ for all k, and therefore the three loops ending at k (boxes 4, 11, and 15) do indeed terminate.//

Corollary: Algorithm (99) computes z of operation (90) with $f = \Sigma_{i=1}^{k} c_i f_i$.

Proof: (99) is a proper subset of (100), namely boxes 2 and 8-12 are missing and box 7 connects directly to box 13. The k of (99) can serve as the required k_h for all h. With this choice of k_h the processing of (100) reduces to the processing of (99) since box 2 is now a null operation and the test at 8 sends control from box 7 directly to box 13. The proof of (100), being independent of the choice of k_h, therefore also proves (99).//

Algorithm (106), for computing the product of asymptotic series, uses lemma 13, i.e., (56) below is computed. A review of the notation is in order. Each f_i comprising the product is of the form

$$f_i = \Sigma_{r=1}^{\infty} a_{ij_{ir}} \varphi_{j_{ir}} \quad \text{with} \quad a_{ij_{ir}} \neq 0 \quad \text{for all} \quad r. \qquad (*5)$$

$\mathrm{ASIND}(f_i;r) = j_{ir}$ and $\mathrm{ASCOF}(f_i;r) = a_{ij_{ir}}$. For $k \geq 1$ the product series is given by

$$\Pi_{i=1}^{k} f_i = \sum_{\text{all } [\ell_1, \ell_2, \ldots, \ell_k]} (a_{1\ell_1} a_{2\ell_2} \cdots a_{k\ell_k} \varphi_{\ell_1} \varphi_{\ell_2} \cdots \varphi_{\ell_k})$$

$$= \Sigma_{t=0}\, p_t \psi_t; \qquad (56), (*6)$$

i.e., $a_{i\ell_i} \neq 0$ if and only if $\ell_i = j_{ir}$ for some r.

The $p_t\psi_t$ notation expresses the use of the distributive law to collect like ψ_t terms. For fixed t, the aim is to find all possible products of k φ's, one from each f_i, that produce ψ_t, i.e., to find all k-tuples $[\ell_1, \ell_2, \ldots, \ell_k]$ such that $\varphi_1\varphi_2\ldots\varphi_k = \psi_t$. Since by (101) there are $m_{t+1} - m_t$ such k-tuples,

$$p_t = \Sigma_{c=m_t}^{m_{t+1}-1} a_c \qquad (*7)$$

where

$$a_c = \Pi_{i=1}^{k} a_{i\ell_i} \qquad (*8)$$

with the ℓ_i from the cth k-tuple, as described in (56).

The k-tuples are given (generated), consistent with the g-ordering, by $q_i(m)$ which denotes the ith entry in the mth k-tuple, i.e., $a_c = \Pi_{i=1}^{k} a_{iq_i(c)}$ in (*8). If some $\varphi_{q_i(m)}$, i.e., φ_{ℓ_i}, does not appear in f_i, then the a_c for the mth k-tuple is zero.

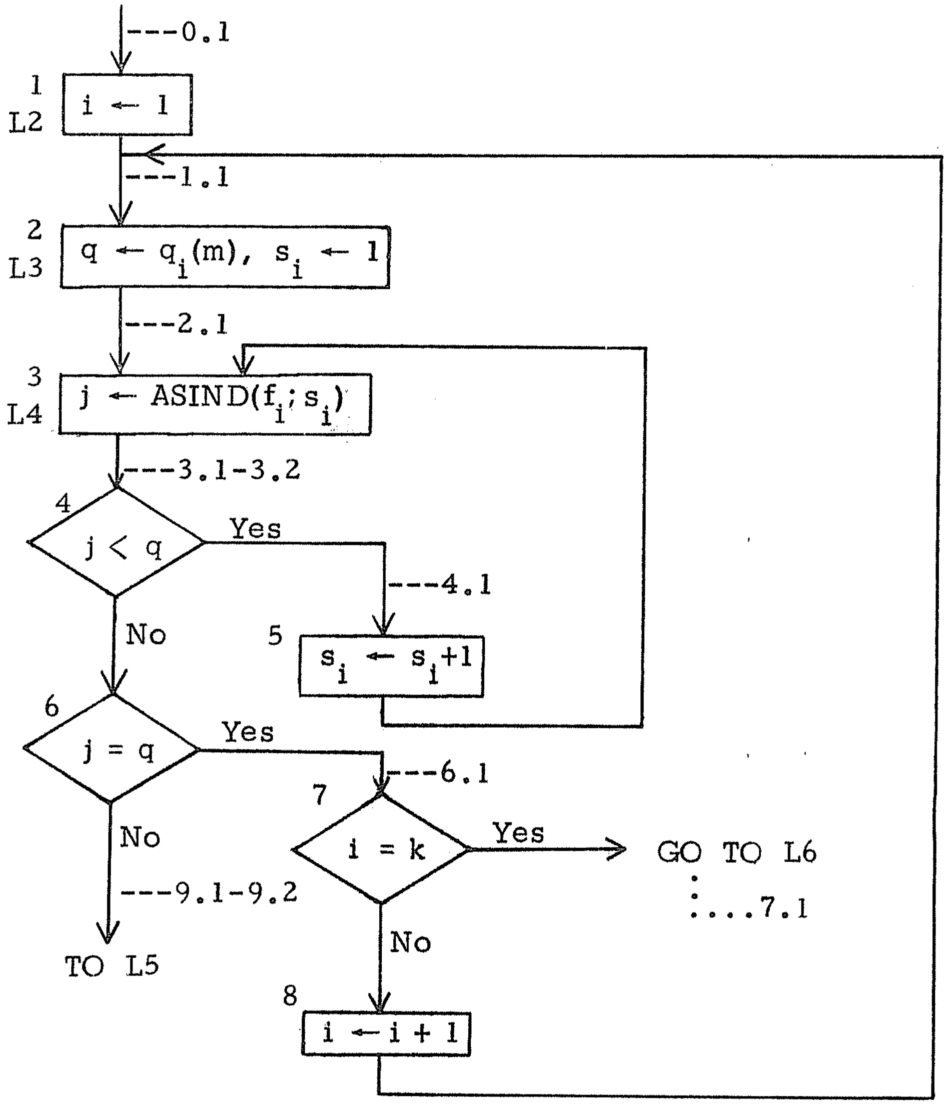
In (106) it will be necessary to remember where φ_{ℓ_i} is located in f_i. The quantity s_i is used in (106) with the meaning that φ_{ℓ_i} is the s_ith term in f_i, i.e., $\ell_i = j_{is_i}$. An assertion is needed saying that certain s_i's are properly set. Let C(d) mean that φ_{ℓ_u} is the s_uth term in f_u for $u = 1, 2, \ldots, d$.

Theorem: Algorithm (106) computes z of operation (90) with $f = \Pi_{i=1}^{k} f_i$.

It is convenient to prove first a section of (106) as the following:

Lemma: Let m be fixed. The code in (106), from line (L2) up to but not including (L5), given in the next flowchart, either (i) computes the s_i ($i = 1,2,\ldots,k$) for the mth k-tuple such that $C(k)$ holds, and exits to (L6) from test 7; or else (ii) shows that $a_m = 0$, and exits to (L5) from test 6.

Proof:



Flowchart for part of Algorithm (106)

Below appear the assertion numbers, the actual assertions and the reasons why each assertion is true.

0.1: $n \geq 1$: Assumption on n.

1.1: $C(i-1)$: From 1: $i = 1$ so $C(0)$ is a null statement. From 6.1: By 6.1 with change in i at 8.

2.1: $q = \ell_i$ of the mth k-tuple: By definition of $q_i(m)$.

3.1: $j = \text{ASIND}(f_i;s_i)$: By 3.

3.2: $q > \text{ASIND}(f_i;s_u)\,(u = 1,2,\ldots,i-1)$: From 2.1: $i = 1$ so range of u is null. From 4.1: By 4.1 with change in s_i at 5.

4.1: $q > \text{ASIND}(f_i;s_u)\,(u = 1,2,\ldots,i)$: By 3.2, 3.1, and the test at 4.

6.1: $C(i)$: Assertion 1.1 holds at 6.1 since $s_1, s_2, \ldots, s_{i-1}$ are unaltered from 1.1. By the test at 6, s_i is set so $C(i)$ holds.

7.1: $C(k)$: By 6.1 with $i = k$.

9.1: $\text{ASIND}(f_i;s_i) > q$: By the tests 4 $(j \geq q)$ and $6(j \neq q)$.

9.2: $a_m = 0$: By 3.2 and 9.1; i.e., the index j_i does not appear in f_i, whence φ_{ℓ_i} does not appear in f_i and so $a_m = 0$.

Possibility (i) is assertion 7.1 and possibility (ii) is assertion 9.2.

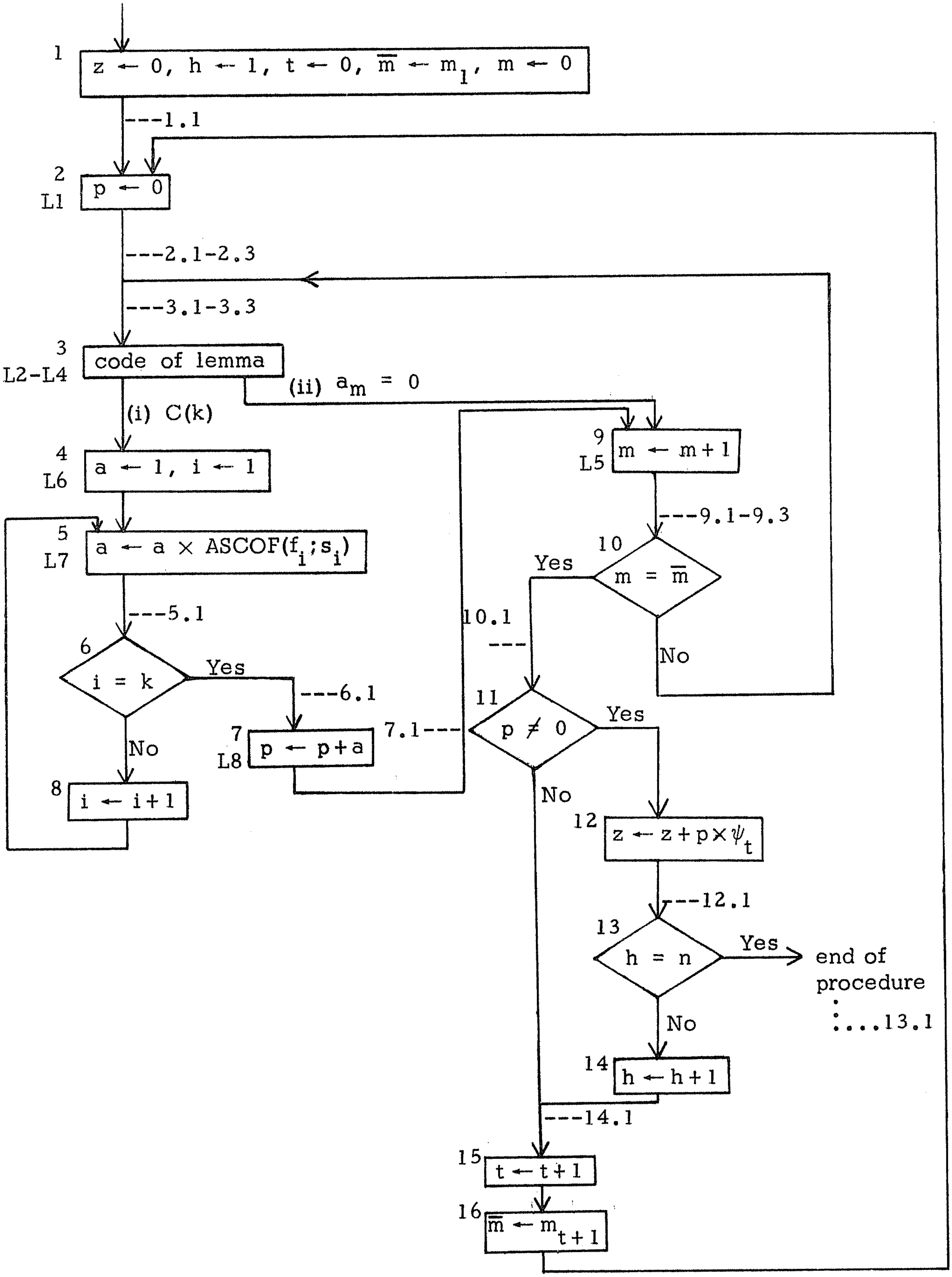
This section of code terminates since the two sourses of potential unending loops both terminate. First, the loop from boxes 3 to 5 to 3 terminates, since q is fixed and j is increasing, whence $j < q$ is not possible indefinitely. Second, the loop from boxes 3 to 6 to 8 to 3 terminates, since i starts at 1, is stepped by 1 after testing for $i = k \geq 1$, and therefore i must eventually equal k.//

Lemma: Variables in (106) are changed in value only as follows:

Variable	Changed at Box Number
a	4, 5
a_m	defined at assertion 6.1
h	1, 14
i	3, 4, 8
j	3
k	--
m	1, 9
m_t	--
$\overline{m}$	1, 16
n	--
p	2, 7
p_t	defined at assertion 10.1
q	3
s_i	3
t	1, 15
z	1, 12

Proof: Inspection of the flowchart.//

Proof of theorem: Use the A(s) notation as defined for the proof of (91) but with ψ and p instead of φ and a. Also note that $m_0 = 0$. Box 3 of the complete flowchart for (106) represents the flowchart of the lemma. The two exits from 3 are labelled to show the two separate results.



Flowchart for Algorithm (106)

Below appear the assertion numbers, the actual assertions and the reasons why each assertion is true.

1.1: $z = A(0) = 0$: By 1.

2.1: $m = m_t$: From 1.1: $t = 0$ and $m_0 = 0 = m$. From 14.1: By 2.3 $\overline{m} = m_{t+1}$. $m = \overline{m}$ at 10; so $m = m_{t+1}$. After the change in t at 15, $m = m_t$.

2.2: $z = A(h-1)$: From 1.1: By 1 and 1.1. From 14.1: By 14.1.

2.3: $\overline{m} = m_{t+1}$: From 1.1: $t = 0$ and $\overline{m} = m_1$ by 1. From 14.1: By 16.

3.1: $p = \Sigma_{c=m_t}^{m-1} a_c$: From 2.1: By 2 and 2.1 giving null sum. From 9.1: By 9.1.

3.2: $z = A(h-1)$: From 2.1: By 2.2. From 9.1: By 9.2.

3.3: $m < \overline{m} = m_{t+1}$: From 2.1: By 2.1, 2.3, and since m_t is a strictly increasing sequence, i.e., $m = m_t < m_{t+1} = \overline{m}$. From 9.1: By 9.3, and $m \neq \overline{m}$ at 10.

5.1: $a = \Pi_{b=1}^{i}$ ASCOF$(f_b;s_b)$: From 4: C(k), 4, and 5. From 8: By 5.1, with change in i at 8, by 5, and by C(k).

6.1: $a = \Pi_{b=1}^{k}$ ASCOF$(f_b;s_b)$, define $a = a_m$ by (*8): By 5.1 with $i = k$.

7.1: $p = \Sigma_{c=m_t}^{m} a_c$: By 3.1, 6.1, and 7.

9.1: $p = \Sigma_{c=m_t}^{m-1} a_c$: From 7.1: By 7.1 and change in m at 9. From 3.1: By 3.1, $a_m = 0$, and 9.

9.2: $z = A(h-1)$: By 3.2 (from either 7.1 or 3.1).

9.3: $m \leq \overline{m} = m_{t+1}$: By 3.3 and 9 (from either 7.1 or 3.1).

10.1: $p = \Sigma_{c=m_t}^{m_{t+1}-1} a_c$, define $p = p_t$ by (*7): By 9.1, 10, and 9.3.

12.1: $z = A(h)$: By 9.2, 10.1, 11, and 12.

13.1: $z = A(n)$: By 12.1 with $h = n$ at 13.

14.1: $z = A(h-1)$: From 12.1: By 12.1 and change in h at 14.

From 10.1: By 9.2 and 11.

Termination for (106) follows as in (91). Note that there is no unending loop from boxes 3 to 9 to 10 to 3 since at 9.3, $m \leq \overline{m}$ and m, changed only at 9, is increasing.//

Algorithm (112) computes $f = g^r$, where r is a real number, by using lemma 15 as discussed on pp. 38-40 of [1] and in the Appendix. The proof of (112) uses the same techniques as the previous five proofs, but the organization of the proof is different. The code for (112) is broken into six sections, named A, B, ..., F. Each section, effectively a subroutine, is proved, by a lemma, to accomplish a stated computation. The six sections are then tied together in a flowchart named G which represents the entire algorithm (112) and which is proved using the lemmas as the main tools. Boxes and assertions are numbered as before with the corresponding letter A-G prefixed. We start with the table of variable-value changes.

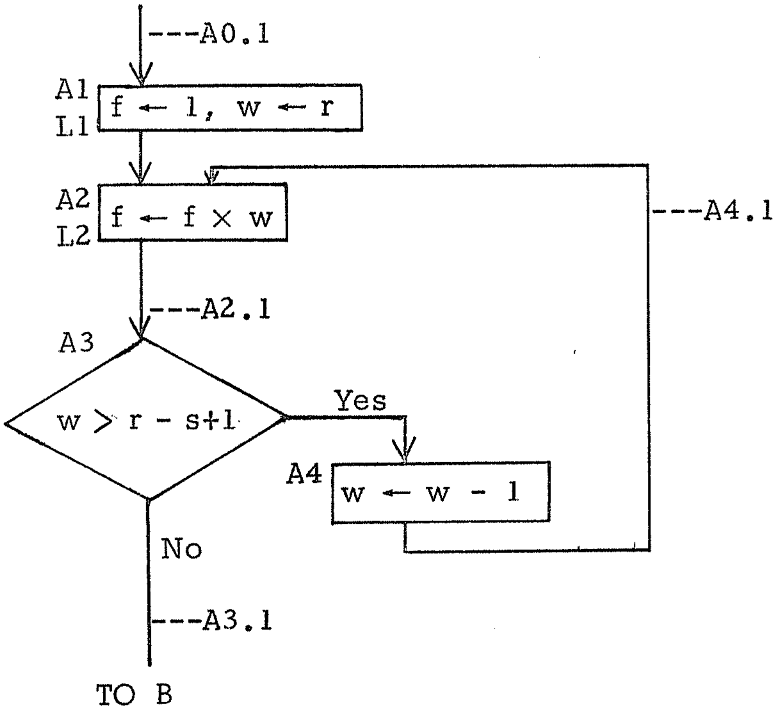
Lemma: Variables in (112) are changed in value only as follows: (A "sub" after a variable denotes a subscripted variable referenced by more than one subscript expression.)

Variable	Changed at Box Number
a_0	G2
b	C4
f	A1, A2
h	B1, B4, C1, C10, D5, D11, E5, E8, F1, F10
i	C5, C8, D1, D7, F1, F8, F11
j	F6, G2
j_h	C2
k	G2, G10, G13
ℓ	D5, D8, F12, G2
m_{sub}	B2, B5, E3, E9
n	--
p_{sub}	D3, D8, D9, D12, F10
q	C1, C6
r	--
s	G2, G10, G13
t	G6
t_0	G2
t_{sub}	D8, D9, D12, F10
$\bar{t}_{sub}$	--
w	A1, A4
z	F5, G1, G2, G12

Proof: Inspection of the flowchart.//

Lemma A: If $s \geq 1$, then section A computes $f = \Pi_{v=0}^{s-1}(r-v)$, i.e., assertion A3.1.

Proof:



Flowchart for Section A of (112)

A0.1: $s \geq 1$: Assumption.

A2.1: $f = \Pi_{v=0}^{r-w}(r-v)$: From A1: By A1 and A2, $w = r$, so $f = r$.

From A4.1: By A4.1 and A2.

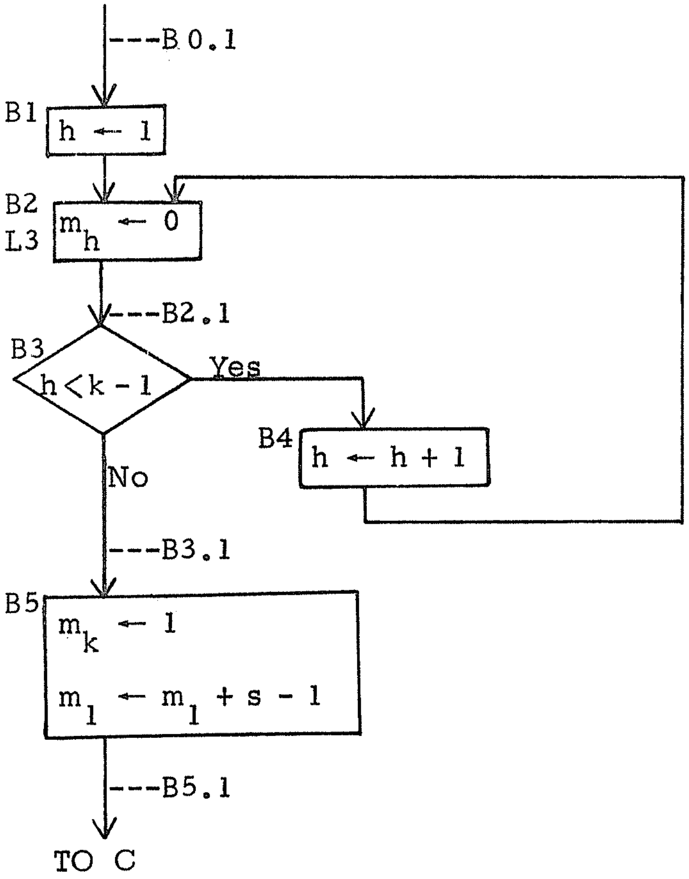
A3.1: $f = \Pi_{v=0}^{s-1}(r-v)$: By A2.1 with $w = r - s + 1$.

A4.1: $f = \Pi_{v=0}^{r-(w+1)}(r-v)$: By A2.1 and A4.

Termination: $s \geq 1$ implies $r \geq r - s + 1$. Hence w, initially r and decremented by 1, must eventually equal the fixed $r - s + 1$.//

Lemma B: For fixed k and s with $k \geq 1$, section B computes the initial k-tuple, $[m_1, m_2, \ldots, m_k]$, as given by assertion B5.1.

Proof:



Flowchart for Section B of (112)

B0.1: $k \geq 1$: Assumption.

B2.1: $m_v = 0$ for $v = 1, \ldots, h$: From B1: By B1, $h = 1$. From B4: By B2.1, B4, and B2.

B3.1: $m_v = 0$ for $v = 1, \ldots, k-1$: By B2.1 with $h = k - 1$.

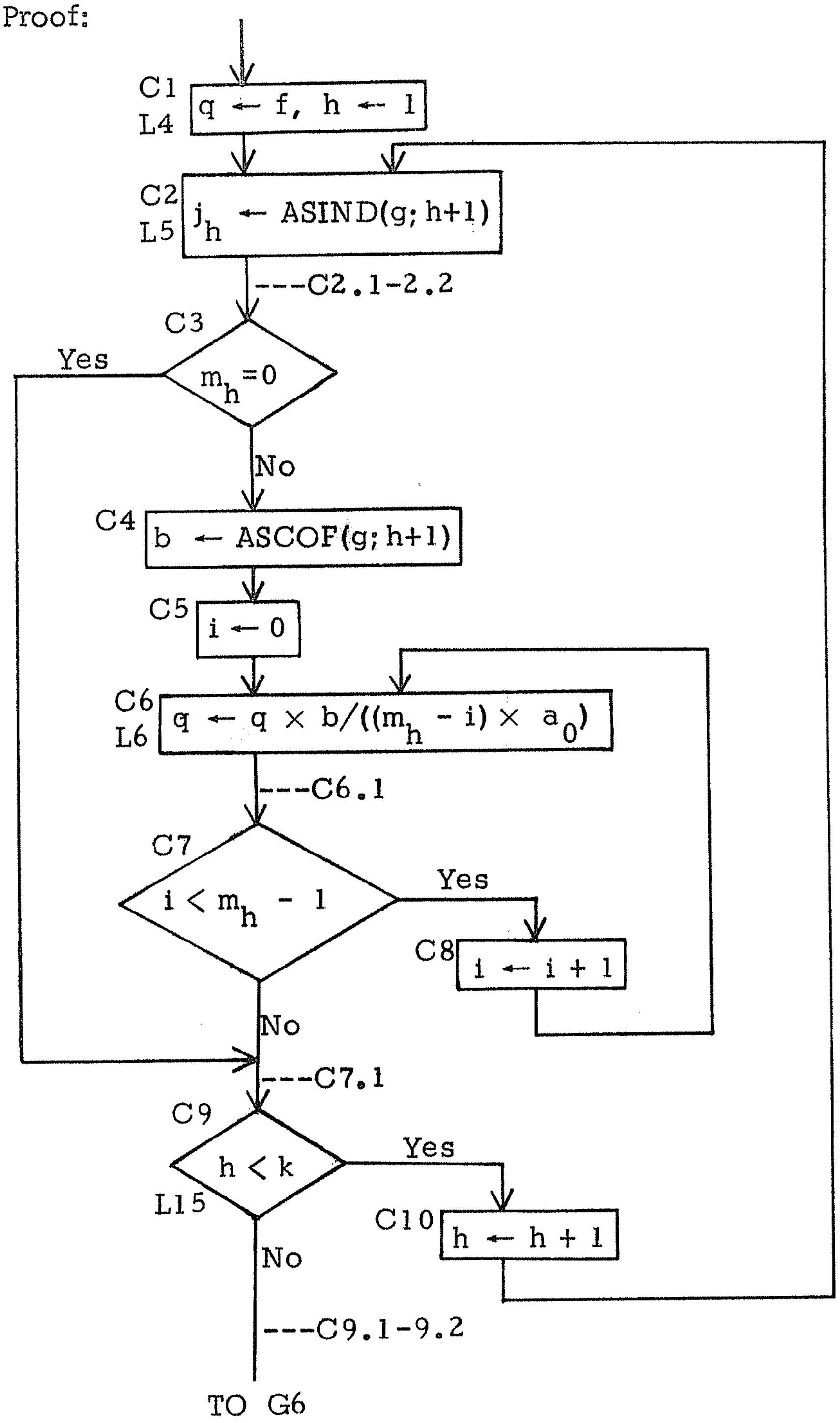
B5.1: $m_1 = s$ if $k = 1$,

$m_1 = s - 1$, $m_2 = \ldots = m_{k-1} = 0$, $m_k = 1$ if $k > 1$: By B3.1 and B5 for the two cases on k, i.e., $m_1 = 1$ or $m_1 = 0$.

Termination: If $h = 1 \geq k - 1$, then B terminates. Otherwise $h < k - 1$ and by B4 and B3, eventually $h \geq k - 1$.//

Lemma C: For fixed $a_0 \neq 0$, f, $k \geq 1$, $s \geq 1$, and the k-tuple $[m_1, m_2, \ldots, m_k]$ with non-negative elements, section C computes q, namely formula (66), and also h_v for $v = 1, 2, \ldots, k$.

Proof:



Flowchart for Section C of (112)

C2.1: $q = Y(h-1) = \frac{f}{m_1!m_2!\ldots m_{h-1}!}\left(\frac{a_{j_1}}{a_0}\right)^{m_1}\left(\frac{a_{j_2}}{a_0}\right)^{m_2}\cdots\left(\frac{a_{j_{h-1}}}{a_0}\right)^{m_{h-1}}$: From C1: By C1, $h = 1$, and hence $q = f$ as required. From C10: By C7.1 and C10.

C2.2: h_v set for $v = 1,\ldots h$: From C1: By C2 and $h = 1$. From C10: By C2.2, C10, and C2.

C6.1: $q = \frac{Y(h-1)}{m_h(m_h-1)\cdots(m_h-i)}\left(\frac{a_{j_h}}{a_0}\right)^{i+1}$: From C5: By C3-C5, $m_h \neq 0$, $b = a_{j_h}$, and $i = 0$; hence by C2.1. By C3 and C5, $m_h - i \neq 0$. From C8: By C6.1, C8, and C6. By C7 and C8, $m_h - i \geq 1$.

C7.1: $q = Y(h)$: From C3 directly: Since $m_h = 0$, m_h contributes $\left(\frac{a_{j_h}}{a_0}\right)^0 = 1$ and $0! = 1$, to C2.1. From C7: By C6.1 with $i = m_h - 1$.

C9.1: $q = Y(k)$: By C7.1 and C9, i.e., $h = k$. $Y(k)$ is the expression (66).

C9.2: h_v set for $v = 1,\ldots,k$: By C2.2 with $h = k$ by C9.

Termination: The outer loop on h: By C1, h starts at 1 and by C10 is increased. Since $k \geq 1$ is fixed, $h \geq k$ eventually. The inner loop on i: By C3, the fixed $m_h \geq 1$, by C5 i starts at 0, and by C8 i is increased; hence $i \geq m_h - 1$ eventually.//

Let $M(\ell)$ denote

t_v for $v=1,2,\ldots,\ell$ exist,

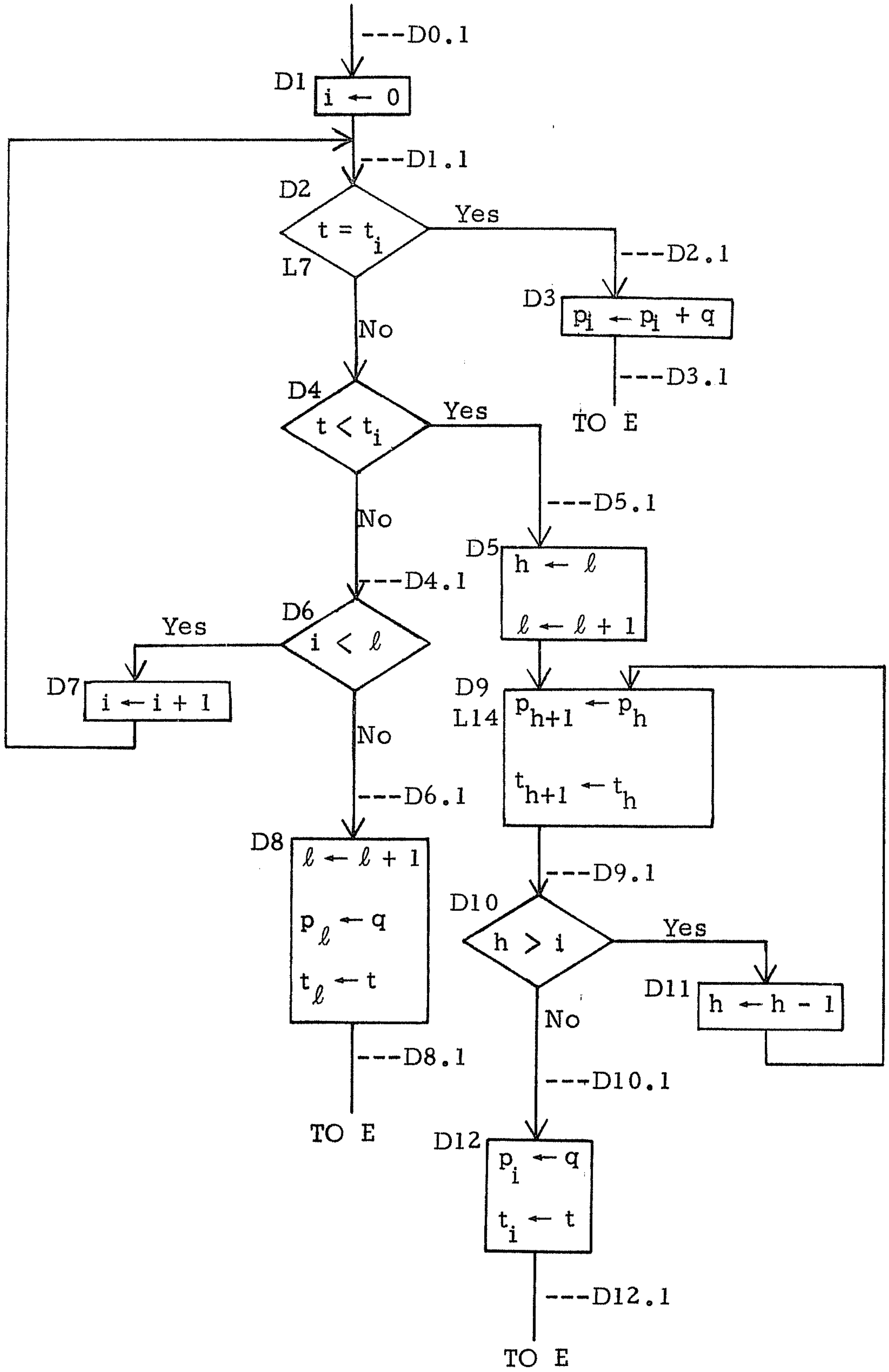
$t_0 = 0$ and $t_{v-1} < t_v$ for $v=1,2,\ldots,\ell$,

p_v for $v = 1,2,\ldots,\ell$ exist,

p_v is the sum of all terms of the form (66) with $t = t_v$ which have been computed but not yet included in z.

Lemma D: If $t > t_0$, if $\ell \geq 0$, if $M(\ell)$ holds except for q, and if t = index of q; then after section D, $\ell \geq 1$ and $M(\ell)$ holds, i.e., q has been added to the proper p_v.

Proof:



Flowchart for Section D of (112)

D0.1: $M(\ell)$ except for q: Assumption.

D1.1: $t > t_v$ for $v = 0,1,2,\ldots,i-1$: From D1: null statement. From D7: By D4.1 and D7.

D2.1: $t > t_v$ for $v = 0,1,2,\ldots,i-1$ and $t = t_i$: By D1.1 and D2.

D3.1: $M(\ell)$: By D2.1 and D3.

D4.1: $t > t_v$ for $v = 0,1,2,\ldots,i$: By D1.1, D2, and D4.

D5.1: $t > t_v$ for $v = 0,1,2,\ldots,i-1$ and $t < t_i$: By D1.1 and D4.

D6.1: $t > t_v$ for $v = 0,1,2,\ldots,\ell$: By D4.1 with $i = \ell$ by D6.

D8.1: $M(\ell)$: By D0.1, D6.1, and D8.

D9.1: p_v and t_v each become p_{v+1} and t_{v+1} respectively for $v = h,\ldots,\ell-1$: From D5: By D9 since $h = \ell - 1$. From D11: By D9.1, D11, and D9.

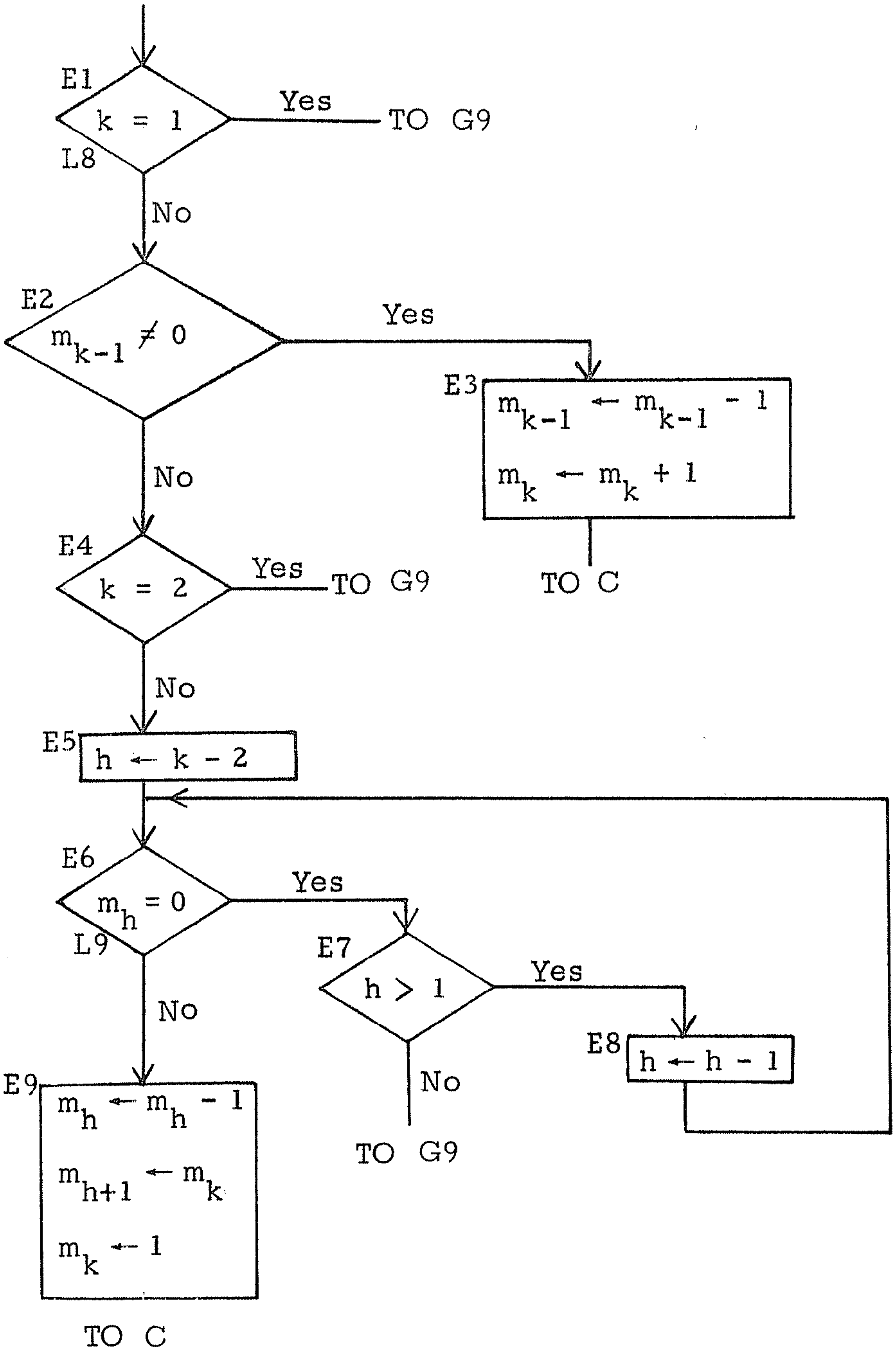
D10.1: p_v and t_v each become p_{v+1} and t_{v+1} respectively for $v = i,\ldots,\ell-1$: By D9.1 with $h = i$.

D12.1: $M(\ell)$: By D5.1, D10.1, and D12.

The three exits from section D (at D3.1, D12.1, and D8.1) correspond to the three cases (i) $t = t_v$, for some v, and q is added to p_v; (ii) t is a new index and appears just before t_v, in which case all the t's and p's, v and larger, are shifted to make room for t and q; and (iii) t is a new index and larger than t_ℓ (including the case $\ell = 0$), in which case t and q are placed at the end. If exit is from D12.1 or D8.1, then $\ell \geq 1$ by D5 or D8. If exit is from D3.1, then the input ℓ is unchanged. Furthermore, on input, $\ell \geq 1$, since, if $\ell = 0$, the exit is from D8.1, because $t > t_0$.

Termination: The loop on i: By D1, i starts at 0 and is increased at D7. Since $\ell \geq 0$ is fixed, $i \geq \ell$ eventually. The loop on h: By D6 and D7, $i \leq \ell$ at D5.1. By D5, h starts at $\ell \geq i$ and is decremented at D11. Since i is fixed in the h loop, $h \leq i$ eventually.//

Lemma E: For fixed $k \geq 1$, $s \geq 1$, and the k-tuple $[m_1, m_2, \ldots, m_k]$ with non-negative elements, section E computes, according to (109), the next k-tuple, if it exists, for k and s, and goes to L4. If there is no further k-tuple, control goes to L10.



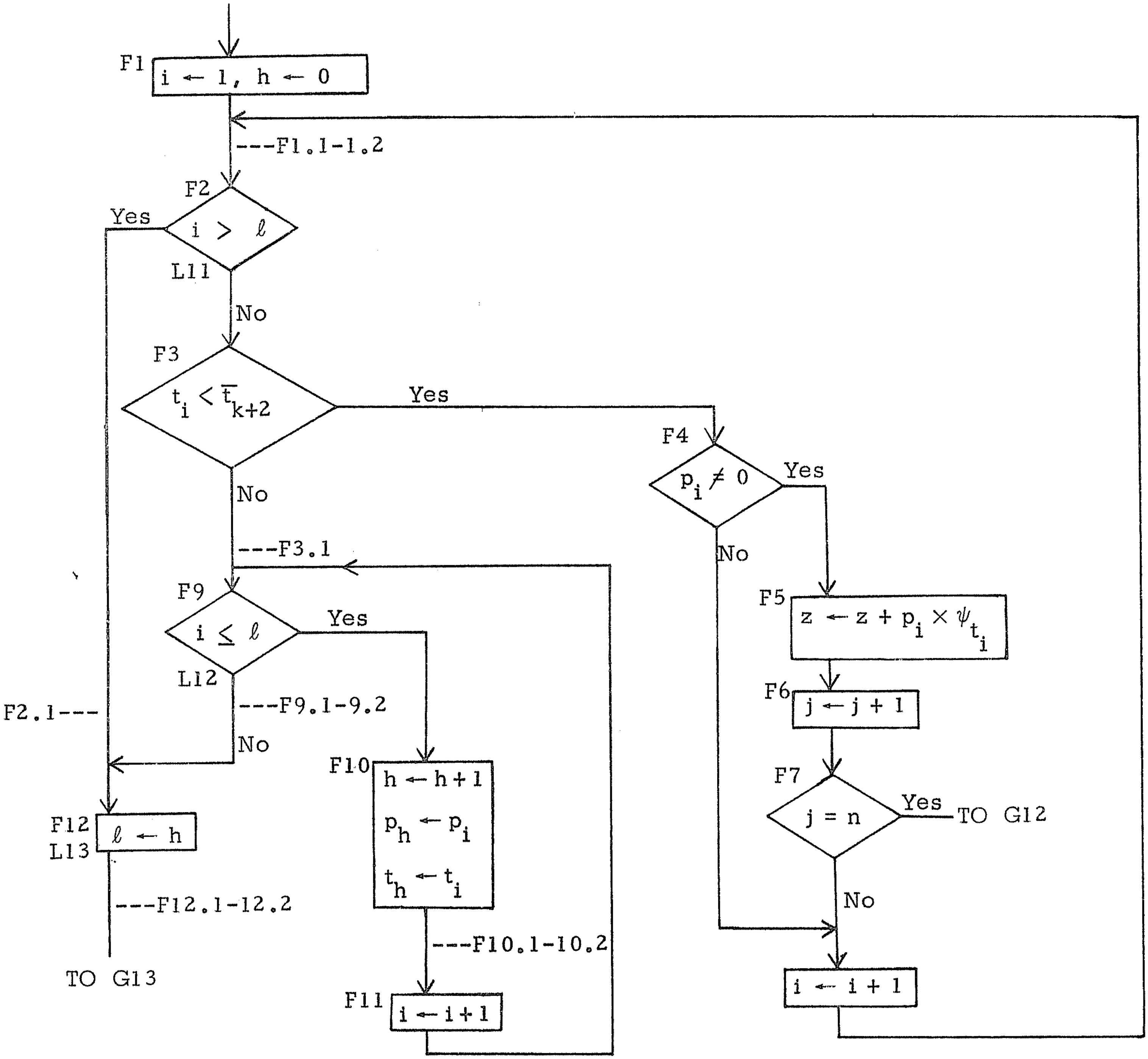
Flowchart for Section E of (112)

Proof: If $k = 1$, no further k-tuple for k and s exists. If $m_{k-1} \neq 0$ then the next k-tuple is given by E3. If $m_{k-1} = 0$ and $k = 2$, then no further k-tuple exists. Hence, after E5, $h \geq 1$. The h loop seeks the first non-zero m_h, $h \leq k - 2$, starting from m_{k-2} (m_{k-1} is already zero by E2). If found, E9 gives the next k-tuple. Note that $h + 1 \neq k$. If no non-zero element found, then no further k-tuple exists. All computed k-tuples have only non-negative elements.

Termination: After E5, $j \geq 1$ and j is decremented at E8. Hence eventually $h \leq 1$.//

Lemma F: Assume z has j terms counting $a_0\psi_0$. Assume $\ell \geq 1$ and $M(\ell)$. Section F adds to z all the p_v such that $t_v < \bar{t}_{k+2}$. All remaining pairs p_v, t_v are compacted (moved up) so that p_v, t_v $(v = 1,2,\ldots,\ell)$ are the terms not added to z. j and ℓ are updated, so that, on exit, z again has h terms, $M(\ell)$ holds, and $\ell \geq 1$.

Proof:



Flowchart for Section F of (112)

F1.1: $p_v \psi t_v$ $(v = 1,2,\ldots,i-1)$ added to z: From F1: By F1, $i = 0$, so range of v is null. From F8: By F1.1, F3, F4, F5, and F8.

F1.2: z has j terms: From F1: Assumption. From F8: By F5 and F6.

F2.1: z has j terms: By F1.2 (no change).

F3.1: Define $i_0 = i \leq \ell$, p_v $(v = i_0,\ldots,\ell)$ should not be added yet to z: By F3 and definition of $\mathcal{T}_{k+2}$. By F2, $i \leq \ell$.

Let $N(c)$ denote that p_v, t_v, for $v = i_0, i_0 + 1,\ldots,c$ have been moved up to p_x, t_x, for $x = 1,2,\ldots,h = c - i_0 + 1$, respectively.

F9.1: $N(\ell)$, $i = \ell + 1$: By F10.1, F9, and F11.

F9.2: $h = \ell - i_0 + 1$: By F9.1, and F10.2 with F11 (here $h = i - i_0$).

F10.1: $N(i)$: From F3.1: By F3.1, F1, and F10, $h = 1$, so p_{i_0}, t_{i_0} are moved to p_1, t_1, respectively. From F11: By F10.1 and F10.

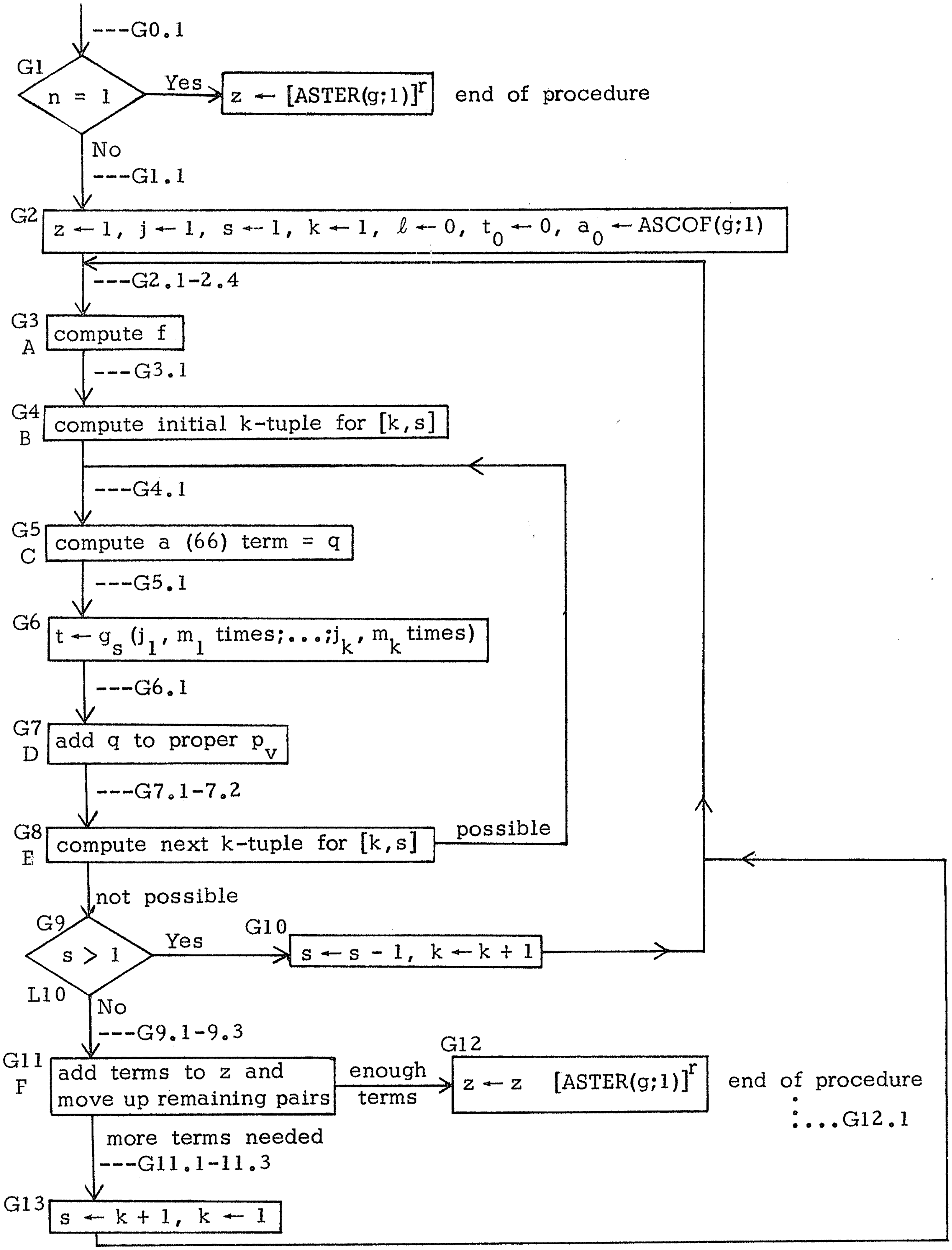
F10.2: $h = i - i_0 + 1$: From F3.1: By F1 and F10, $h = 1$, and by F3.1, $i = i_0$. From F10.1: By F10.2, F11, and F10.

F12.1: z has j terms: From F2.1: By F2.1. From F9.1: unchanged from F1.2.

F12.2: $M(\ell)$ and $\ell \geq 0$: From F2.1: By F12, $\ell = h$, and $h = 0$ from F1. From F9.1: by F9.2, $h = \ell - i_0 + 1$, the proper value since $i_0 \leq \ell$, $\ell \geq 1$.

Termination: The loop in i through F4: By F1, i starts at 1 and is incremented at F8. Since $\ell \geq 1$ is fixed, $i > \ell$ eventually. The loop on i through F11: By F2, $i \leq \ell$ at F3.1, and i is incremented at F11: Eventually $i > \ell$.//

Theorem: Algorithm (112) computes $f = g^r$ according to lemma 15 in [1].



Overall Flowchart for Algorithm (112)

Proof:

G0.1: $n \geq 1$: Assumption. Note if $n = 1$, z is correct and (112) exits.

G1.1: $n \geq 2$: By G0.1 and G1.

G2.1: z has j terms counting $a_0\psi_0$: From G2: By G2, $j = 1$. From G10: Unchanged from G2.1, From G13: By G11.1.

G2.2: $s \geq 1$, $k \geq 1$: From G2: By G2. From G10: By G9, $s \geq 2$, and by G10, $s \geq 1$. $k \geq 1$ from G10 and G2.2. From G13: By G13 and G2.2.

G2.3: $M(\ell)$: From G2: $\ell = 0$, so $M(\ell)$ is null. From G10: Unchanged from G7.1. From G13: By G11.2.

G2.4: $\ell \geq 0$: From G2: By G2. From G10: Unchanged from G7.2. From G13: By G11.3.

G3.1: $f = \Pi_{v=0}^{s-1} (r-v)$: Since $s \geq 1$ by G2.2, lemma A.

G4.1: Next k-tuple set for computing a (66) term: From G4: By lemma B since $k \geq 1$ by G2.2. From G8: By G2.2 and lemma E.

G5.1: q = a (66) term: By lemma C since $a_0 \neq 0$ by (64), since $k \geq 1$ and $s \geq 1$ by G2.2, and since k-tuple has non-negative elements by G4.1.

G6.1: t = index of q: By definition of t and by C9.2 for h_i.

G7.1: $M(\ell)$: By lemma D, since $\ell \geq 0$ by G2.4, and since $M(\ell)$ except for q by G2.3. Also, by G2.2, $k \geq 1$, whence there is a non-empty k-tuple with $m_k > 0$; and so in G6, we get $t > t_0 = 0$ (since $\psi_0 = 1$, corresponding to $k = 0$ and empty k-tuple).

G7.2: $\ell \geq 1$: By lemma D.

G7.3: $k \geq 1$: By G2.2.

G9.1: $s = 1$: By G9 and G2.2.

G9.2: z has j terms counting $a_0\psi_0$: Unchanged from G2.1.

G9.3: $\ell \geq 1$ and $M(\ell)$: Unchanged from G7.1-7.2.

G11.1: z has j terms counting $a_0\psi_0$: By lemma F and G9.3.

G11.2: $M(\ell)$: By Lemma F.

G11.3: $\ell \geq 1$: By Lemma F.

G12.1: $z = g^r$: z has j terms, by Lemma F, and $j = n$ by meaning of "enough terms." At G12 include overall factor $a_0^r \varphi_0^r$ and halt.

Termination: Each box A-F terminates by the appropriate lemma. The other boxes clearly terminate.

The loop from G5 to G8 to G5: There is only a finite number of k-tuples for fixed $[k, s]$.

The loop from G3 to G9 to G3: s is fixed at G3 and s is only changed at G10. Hence eventually $s = 1$ and exit at G9.

The loop through G13: Assuming non-infinite set of zero terms p_i that can be added by virtue of $\overline{t}_{k+2}$ (cf. (91)), control exit through G12.//

APPENDIX

We list here the equations and algorithms of [1] which are referred to in the present report. Any discrepancies, between the material in [1] and that below, arise from corrections made after the earlier report was published, many of them discovered in the process of proving the algorithms. What we have here is a minimum of information, to enable the reader to follow our discussion without reference to [1]. However, for all details of definitions, proofs, etc., the reader must go to the first report [1].

A sequence of functions $\varphi_0, \varphi_1, \varphi_2, \dots$ is said to be an asymptotic sequence for $x \to \alpha$ in $\mathscr{D}$, where $\mathscr{D}$ is the intersection of the domains of definition of all the functions and α is in the closure of $\mathscr{D}$, if $\varphi_k(x)/\varphi_{k-1}(x) \to 0$ as $x \to \alpha$ in $\mathscr{D}$, i.e.,

$$\varphi_k = \underline{o}(\varphi_{k-1})_\alpha \quad \text{for} \quad k = 1,2,3,\dots. \tag{36}$$

We write

$$f \approx \Sigma_{k=0}^{\infty} a_k \varphi_k, \tag{*9}$$

and say that the right-hand series is an asymptotic series for the function f, if $[\varphi_k]_{k=0}^{\infty}$ is an asymptotic sequence and, when

$$S_m = \Sigma_{k=0}^{m-1} a_k \varphi_k, \tag{40}$$

we have

$$f - S_m = a_m \varphi_m + \underline{o}(\varphi_m)_\alpha, \tag{43}$$

for $m = 0,1,2,\dots.$ Then we have, for all m,

$$a_m = \lim_{x \to \alpha} \{[f(x) - S_m(x)]/\varphi_m(x)\}. \tag{44}$$

If

$$f_i(x) \approx \Sigma_{j=0}^{\infty} a_{ij}\varphi_j(x) \tag{*10}$$

for $i = 1,2,\ldots,k,$ as $x \to \alpha$ in $\mathfrak{F}$, the intersection of $\mathfrak{D}$ with the domains of all the f_i; then for any $c_1, c_2, \ldots, c_k$,

$$\Sigma_{i=1}^{k} c_i f_i(x) \approx \Sigma_{j=0}^{\infty} (\Sigma_{i=1}^{k} c_i a_{ij}) \varphi_j(x). \tag{50}$$

Further, if (*10) holds uniformly in i, for $i = 1,2,3,\cdots$; then

$$\Sigma_{i=1}^{\infty} c_i f_i(x) \approx \Sigma_{j=0}^{\infty} (\Sigma_{i=1}^{\infty} c_i a_{ij}) \varphi_j(x), \tag{51}$$

so long as $\Sigma_{i=1}^{\infty} c_i$ is absolutely convergent and $\Sigma_{i=1}^{\infty} c_i a_{ij}$ converges for $j = 0,1,2,\cdots.$

If (*10) holds and if the set of functions $\Pi_{i=1}^{k} \varphi_{j_i}$, with each j_i ranging through $0,1,2,\ldots,$ can be rearranged into an asymptotic sequence $[\psi_t]_{t=0}^{\infty}$ (i.e., for each $[j_1, j_2, \cdots, j_k]$ we have a corresponding $t = g_k(j_1, j_2, \cdots, j_k)$); then

$$\Pi_{i=1}^{k} f_i \approx \Sigma_{t=0}^{\infty} p_t \psi_t, \tag{56}$$

where p_t is the sum of all products $a_{1j_1} a_{2j_2} \cdots a_{kj_k}$ for which $t = g(j_1, j_2, \cdots, j_k)$; i.e.,

$$p_t = \Sigma_{j_1=0}^{\infty} \Sigma_{j_2=0}^{\infty} \cdots \Sigma_{j_k=0}^{\infty} a_{1j_1} a_{2j_2} \cdots a_{kj_k} \delta_{t\, g(j_1, j_2, \cdots, j_k)}. \tag{*11}$$

If the set of all products $\Pi_{i=1}^{k} \varphi_{j_i}$, with not only all j_i ranging through $0,1,2,\cdots,$ but also k ranging through $1,2,3,\cdots,$ can be ordered in an asymptotic sequence $[\psi_t]_{t=0}^{\infty}$; and further if, for every j_1 and j_2 with $j_1 < j_2$, there is a power k such that $\varphi_{j_1}^{k} = \underline{o}(\varphi_{j_2})$, while for every j_1 and k, there is a j_2 such that $\varphi_{j_2} = \underline{o}(\varphi_{j_1}^{k})$; then we say that $[\varphi_j]_{j=0}^{\infty}$ is a <u>totally multiplicative</u> sequence.

If $[\varphi_j/\varphi_0]_{j=0}^{\infty}$ is a totally multiplicative asymptotic sequence and if (*9) holds, with $a_0 \neq 0$; then, for any real r,

$$f^r \approx a_0{}^r \varphi_0{}^r \Sigma_{t=0}^{\infty} p_t \psi_t, \tag{65}$$

where p_t is the sum of all terms

$$\frac{r(r-1)\cdots(r-i_1-i_2-\cdots-i_k+1)}{i_1!i_2!\cdots i_k!}\left(\frac{a_1}{a_0}\right)^{i_1}\left(\frac{a_2}{a_0}\right)^{i_2}\cdots\left(\frac{a_k}{a_0}\right)^{i_k}, \tag{66}$$

[<u>Note the correction to (66) from [1]</u>] with each $i_j \geq 0$ for $1 \leq j < k$, but $i_k \geq 1$, and any $k = 0,1,2,\cdots,$ for which

$$g_{i_1+i_2+\cdots+i_k}(1,\ i_1 \text{ times};\ 2,\ i_2 \text{ times};\ \cdots;\ k,\ i_k \text{ times}) = t \tag{*12}$$

(when $k = 0$; $t = 0$, $p_0 = 0$, and $\psi_0 = 1$).

In the algorithms, instructions end with a comma, and are executed serially; "a ← b" means that the variable a is given the value of the expression b; the notation "GO TO label" denotes an unconditional jump (transfer to labelled instruction next); a label, put in parentheses in the left-hand margin, refers to the instruction immediately to its right; the the notation "IF statement {instructions}" means that the instructions in curly brackets are obeyed only if the statement is true, otherwise they are omitted; "end of procedure" is a jump to whatever is to be done after the procedure is completed.

We write

$$z \leftarrow \text{ASYMP } (f;\ x,\ \alpha;\ L;\ n) \tag{78}$$

for the instruction assigning the first n non-zero terms of the asymptotic series for $f(x)$, as $x \to \alpha$, in terms of the asymptotic sequence $[\varphi_k(x)]_{k=0}^{\infty}$ labelled L. If (*9) holds, with

$$\left.\begin{aligned} a_{j_h} &\neq 0 \quad \text{for} \quad h = 1,2,\cdots,n\ (j_1 < j_2 < \cdots < j_n), \\ a_j &= 0 \quad \text{if} \quad j < j_n \text{ and } j \neq j_h\ (h = 1,2,\cdots,n); \end{aligned}\right\} \tag{79}$$

then the asymptotic package should return

$$z = \Sigma_{h=1}^{n} a_{j_h} \varphi_{j_h}(x) \tag{80}$$

on execution of (78).

We write

$$z \leftarrow \text{LIM } (f;x) \tag{81}$$

for the instruction which returns $z = \lim_{x \to \infty} f(x)$, when such a limit exists. We assume that, by a change of variable, the asymptotic series are made to have the limit $\alpha = +\infty$, and the "α" is omitted from the instruction (78). The instruction (78) then becomes

$$z \leftarrow \text{ASYMP } (f;\ y;\ L;\ n). \tag{90}$$

The formal computation of (80) by (78) can now be written as an algorithm:

```
        $j \leftarrow 0,\ h \leftarrow 1,\ z \leftarrow 0,$
(L1)    $\varphi \leftarrow \varphi_j(y),\ j \leftarrow j+1,\ a \leftarrow \mathrm{LIM}((f-z)/\varphi;\ y),$
        IF  $a \neq 0$  $\{z \leftarrow z + a \times \varphi,$
                  IF  $h = n$  {end of procedure},  $h \leftarrow h+1\},$          (91)
        GO TO L1.
```

We adopt notations, corresponding to (78),

$$a \leftarrow \mathrm{ASCOF}\ (f; x, \alpha; L; h), \tag{92}$$

$$t \leftarrow \mathrm{ASTER}\ (f; x, \alpha; L; h), \tag{93}$$

$$j \leftarrow \mathrm{ASIND}\ (f; x, \alpha; L; h); \tag{94}$$

for instructions which put a_{j_h} for a, $a_{j_h}\ \varphi_{j_h}(x)$ for t, and j_h for j, when (*9) holds with (79) (i.e. $f(x)$ has the expansion (80) in non-zero terms). Then (91) can be written as

```
            $h \leftarrow 1,\ z \leftarrow 0,$
(L1)        $z \leftarrow z + \mathrm{ASTER}(f;\ y;\ L;\ h),$
            IF  $h = n$  {end of procedure},                       (97)
            $h \leftarrow h+1,$ GO TO L1.
```

To effect the summation expressed in (50), we use the following algorithm:

```
        $z \leftarrow 0$, $i \leftarrow 1$,
(L1)    $h_i \leftarrow 1$, $\ell_i \leftarrow$ ASIND$(f_i;y;L;1)$,
        IF $i = k$ {GO TO L2}, $i \leftarrow i+1$, GO TO L1,
(L2)    $h \leftarrow 1$, $j \leftarrow 0$,
(L3)    $a \leftarrow 0$, $i \leftarrow 1$,
(L4)    IF $\ell_i = j$ {$a \leftarrow a + c_i \times$ ASCOF$(f_i;y;L;h_i)$,
                $h_i \leftarrow h_i + 1$, $\ell_i \leftarrow$ ASIND $(f_i;y;L;h_i)$},
        IF $i = k$ {GO TO L5}, $i \leftarrow i+1$, GO TO L4,
(L5)    IF $a \neq 0$ {$z \leftarrow z + a \times \varphi_j(y)$,
                IF $h = n$ {end of procedure},
                $h \leftarrow h+1$}, $j \leftarrow j+1$, GO TO L3.
```
(99)

[Note change of notation from [1]: j_i replaced by ℓ_i.] In general, the infinite summations in (51) cannot be computed without appeal to a limiting process (sometimes computable); but a specially tractable case occurs when, for each index h, there is a known index k_h, such that $f_i \approx \Sigma_{j=h+1}^{\infty} a_{ij}\varphi_j$ for all $i > k_h$ (i.e., the first $h+1$ functions $\varphi_0, \varphi_1, \cdots, \varphi_h$ appear only in the asymptotic expansions of the first k_h functions $f_1, f_2, \cdots, f_{k_h}$) [Note corrections to [1]]. The sums $\Sigma_{i=1}^{\infty} c_i a_{ij}$ appearing in (51) can then be written as $\Sigma_{i=1}^{k_j} c_i a_{ij}$ and computed finitely. To compute (51), we use (99), modified by respectively replacing the first line and the line labelled L3 by

$$\left.\begin{array}{lll} & & z \leftarrow 0,\ i \leftarrow 1,\ k \leftarrow \max(1,k_0), \\ \text{and} & \left\{\begin{array}{l}\text{(L3)} \\ \\ \text{(L6)} \\ \\ \end{array}\right. & \begin{array}{l} a \leftarrow 0,\ i \leftarrow 1,\ \text{IF}\ \ k \geq k_j\ \{\text{GO TO L4}\}, \\ m \leftarrow k+1,\ k \leftarrow k_j, \\ h_m \leftarrow 1,\ \ell_m \leftarrow \text{ASIND}(f_m;y;L;1), \\ \text{IF}\ m = k\ \ \{\text{GO TO L4}\},\ \ m \leftarrow m+1,\ \text{GO TO L6}. \end{array} \end{array}\right\} \tag{100}$$

[Note corrections to [1]].

To compute the product (56), we first establish some notation. We write $\underset{\sim}{q}$ for the k-tuple of indices $[q_i]_{i=1}^{k}$ and suppose that these can be ordered by a single index m in a sequence, determined by the functions $q_i(m)$ $(i=1,2,\cdots,k)$,

$$[\underset{\sim}{q}(m)]_{m=0}^{\infty} = [[q_i(m)]_{i=1}^{k}]_{m=0}^{\infty}, \tag{*13}$$

so that every k-tuple $\underset{\sim}{q}$ has a unique index m, and if $m \leq m'$ then $g(\underset{\sim}{q}(m)) \leq g(\underset{\sim}{q}(m'))$. We define

$$m_t = \min\{m:\ \ g(\underset{\sim}{q}(m)) \geq t\}, \tag{101}$$

so that $g(\underset{\sim}{q}(m)) = t$ for $m = m_t,\ m_t+1,\cdots,m_{t+1}-1,$ and for no other values of m. [Note that, in [1], what is written q here was written j.] We can now compute (56) by the algorithm:

```
        $z \leftarrow 0,\ h \leftarrow 1,\ t \leftarrow 0,\ \overline{m} \leftarrow m_1,\ m \leftarrow 0,$
(L1)    $p \leftarrow 0,$
(L2)    $i \leftarrow 1,$
(L3)    $q \leftarrow q_i(m),\ s_i \leftarrow 1,$
(L4)    $j \leftarrow$ ASIND $(f_i;y;\overline{L};s_i),$
        IF $j < q$ $\{s_i \leftarrow s_i+1,$ GO TO L4$\},$
        IF $j = q$ $\{$IF $i = k$ $\{$GO TO L6$\},$ $i \leftarrow i+1,$ GO TO L3$\},$
(L5)    $m \leftarrow m+1,$ IF $m = \overline{m}$ $\{$IF $p \neq 0$ $\{z \leftarrow z + p \times \psi_t(y),$          (106)
                                  IF $h = n$ $\{$end of procedure$\},$
                                  $h \leftarrow h+1\},\ t \leftarrow t+1,\ \overline{m} \leftarrow m_{t+1},$
                          GO TO L1$\},$ GO TO L2,
(L6)    $a \leftarrow 1,\ i \leftarrow 1,$
(L7)    $a \leftarrow a \times$ ASCOF $(f_i;y;\overline{L};s_i),$
        IF $i = k$ $\{$GO TO L8$\},$ $i \leftarrow i+1,$ GO TO L7,
(L8)    $p \leftarrow p + a,$ GO TO L5.
```

Finally to compute the power of an asymptotic series according to (65), we use the algorithm (112), given below [Note several important corrections to [1]]. The ordering of the k-tuples of indices, which is crucial to the computation, is hierarchically defined as follows. Let the k-tuple be $[i_{j_1}, i_{j_2}, \cdots, i_{j_k}] = [m_1, m_2, \cdots, m_k]$ with all $m_h \geq 0$

and $m_k \geq 1$, where $m_h = i_{j_h}$ is the power of φ_{j_h} occurring in a given term of the expansion, whose coefficient is given in (66); and where the indices j_h correspond to non-zero terms in the expansion of f, as in (79) and (80). [Note: By an unfortunate oversight, in the treatment of algorithm (112) in [1], the notation used was h_j instead of j_h. Here we have interchanged the letters j and h throughout (112) to restore consistency.] Let $\Sigma_{h=1}^{k} m_h = s$. Then, first, all k-tuples are ordered by non-decrease of the value of $k + s$. Secondly, for each constant $k + s$, the order is that of non-decreasing k. Finally, for each fixed pair $[k,s]$, the order is reverse-lexicographic for the "word" $[m_1, m_2, \cdots, m_k]$. This is expressed by saying that, if $[m_1, m_2, \cdots, m_k]$ immediately precedes $[m'_1, m'_2, \cdots, m'_k]$; then either, for some $h = 1, 2, \cdots, k-2$,

$$\left.\begin{array}{ll} & m_h > 0 \text{ and } m_{h+1} = m_{h+2} = \cdots = m_{k-1} = 0, \\ \text{when} & m'_1 = m_1, \cdots, m'_{h-1} = m_{h-1}, \; m'_h = m_h - 1, \; m'_{h+1} = m_k, \\ & m'_{h+2} = \cdots = m'_{k-1} = 0, \quad \text{and} \quad m'_k = 1; \\ \text{or} & m_{k-1} > 0, \\ \text{when} & m'_{k-1} = m_{k-1} - 1, \; m'_k = m_k + 1; \end{array}\right\} \qquad (*14)$$

while if, instead, either $k = 1$ or $m_1 = m_2 = \cdots = m_{k-1} = 0$ (the only other possibilities), then, for the given values of k and s, the k-tuple $[m_1, \cdots, m_k]$ has no successor. [Note: (*14) is a simplified and corrected version of (109) in [1].]

For a given value of $k + s = u$, the least value of $t = g_s(j_1, m_1 \text{ times}; \cdots, j_k, m_k \text{ times})$ is defined to be $\overline{t}_u$. [Compare (110) and (111) of [1].] Then clearly $\overline{t}_u \leq \overline{t}_{u'}$ if $u \leq u'$, and we know that terms with $k + s > u$ cannot contribute to p_t for $t < \overline{t}_u$. This fact is used in the algorithm (between (L11) and (L12)) in determining whether complete terms are collected. [Note considerable number of corrections to [1].]

```
        IF $n = 1$ {$z \leftarrow [\text{ASTER}(g;y;\overline{L};1)]^r$, end of procedure},
        $z \leftarrow 1$, $j \leftarrow 1$, $s \leftarrow 1$, $k \leftarrow 1$, $\ell \leftarrow 0$, $t_0 \leftarrow 0$, $a_0 \leftarrow \text{ASCOF}(g;y;\overline{L};1)$,
(L1)    $f \leftarrow 1$, $w \leftarrow r$,
(L2)    $f \leftarrow f \times w$, IF $w > r - s + 1$ {$w \leftarrow w - 1$, GO TO L2}, $h \leftarrow 1$,
(L3)    $m_h \leftarrow 0$, IF $h < k - 1$ {$h \leftarrow h + 1$, GO TO L3}, $m_k \leftarrow 1$, $m_1 \leftarrow m_1 + s - 1$,
(L4)    $q \leftarrow f$, $h \leftarrow 1$,
(L5)    $j_h \leftarrow \text{ASIND}(g;y;\overline{L};h+1)$,
        IF $m_h = 0$ {GO TO L15}, $b \leftarrow \text{ASCOF}(g;y;\overline{L};h+1)$, $i \leftarrow 0$,
(L6)    $q \leftarrow q \times b/((m_h - i) \times a_0)$, IF $i < m_h - 1$ {$i \leftarrow i + 1$, GO TO L6},
(L15)   IF $h < k$ {$h \leftarrow h + 1$, GO TO L5},
        $t \leftarrow g_s(h_1, m_1 \text{ times}; h_2, m_2 \text{ times}; \cdots; h_k, m_k \text{ times})$, $i \leftarrow 0$,
(L7)    IF $t = t_i$ {$p_i \leftarrow p_i + q$, GO TO L8},
```

(112)

```
        IF $t < t_i$ {$h \leftarrow \ell$, $\ell \leftarrow \ell+1$,
(L14)               $p_{h+1} \leftarrow p_h$, $t_{h+1} \leftarrow t_h$, IF $h > i$ {$h \leftarrow h-1$, GO TO L14},
                    $p_i \leftarrow q$, $t_i \leftarrow t$, GO TO L8},
        IF $i < \ell$ {$i \leftarrow i+1$, GO TO L7}, $\ell \leftarrow \ell+1$, $p_\ell \leftarrow q$, $t_\ell \leftarrow t$,
(L8)    IF $k = 1$ {GO TO L10}, IF $m_{k-1} \neq 0$ {$m_{k-1} \leftarrow m_{k-1} - 1$, $m_k \leftarrow m_k + 1$,
                                                   GO TO L4},
        IF $k = 2$ {GO TO L10}, $h \leftarrow k-2$,
(L9)    IF $m_h = 0$ {IF $h > 1$ {$h \leftarrow h-1$, GO TO L9}, GO TO L10},
        $m_h \leftarrow m_h - 1$, $m_{h+1} \leftarrow m_k$, $m_k \leftarrow 1$, GO TO L4,
(L10)   IF $s > 1$ {$s \leftarrow s-1$, $k \leftarrow k+1$, GO TO L1}, $i \leftarrow 1$, $h \leftarrow 0$,
(L11)   IF $i > \ell$ {GO TO L13},
        IF $t_i < \overline{t}_{k+2}$ {IF $p_i \neq 0$ {$z \leftarrow z + p_i \times \psi_{t_i}(y)$, $j \leftarrow j+1$,
                                  IF $j = n$ {$z \leftarrow z \times [\mathrm{ASTER}(g;y;\overline{L};1)]^r$,
                                          end of procedure}},
                  $i \leftarrow i+1$, GO TO L11},
(L12)   IF $i \leq \ell$ {$h \leftarrow h+1$, $p_h \leftarrow p_i$, $t_h \leftarrow t_i$, $i \leftarrow i+1$, GO TO L12},
(L13)   $\ell \leftarrow h$, $s \leftarrow k+1$, $k \leftarrow 1$, GO TO L1.
```

(112)
(cont.)

References

1. Halton, J. H., Asymptotics for formula-manipulation. Computer Sciences Technical Report No. 54, University of Wisconsin, 1969. Also Proc. of IBM 1968 Summer Institute on Symbolic Manipulation, Tobey, R. G. (Ed.), to appear.

2. Floyd, R. W., Assigning meanings to programs. Proc. of a Symposium in Applied Mathematics, Vol. 19--Mathematical Aspects of Computer Science, Schwartz, J. T. (Ed.), American Mathematical Society, Providence, R. I., 1967, 19-32.

3. Knuth, D. E., The Art of Computer Programming, Vol. 1--Fundamental Algorithms. Addison-Wesley, Reading, Mass., 1968, section 1.2.1.

4. London, R. L., Computer programs can be proved correct. Proc. of the Fourth Systems Symposium: Formal Systems and Non-Numeric Problem Solving by Computers, Case Western Reserve University, to appear.